

ЎЗБЕКИСТОН РЕСПУБЛИКАСИ  
ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ ВАЗИРЛИГИ

ГУЛИСТОН ДАВЛАТ УНИВЕРСИТЕТИ

Ахборот технологиялари кафедраси

## **Дастурлаш асослари**

фанидан ўқув-услубий мажмуа

Билим соҳаси:	Гуманитар
Таълим соҳаси:	130000 – Математика
Таълим йўналиши:	5130100-Математика

Гулистон-2017

Ўқув–услугий мажмуа Олий ва ўрта махсус таълим вазирлиги томонидан 2017 йил 24.08. даги 605 - сонли буйруғининг 2 - иловаси билан тасдиқланган Дастурлаш асослари фани дастури (№ БД -5130100 – 2.01 2017 – йил 18.08) талаблари асосида тайёрланган.

**Тузувчи:** А.А.Қаландаров - ГулДУ “Ахборот технологиялари”  
кафедраси катта ўқитувчиси

**Тақризчи:** С.И.Қулмаматов - ГулДУ “Ахборот технологиялари”  
кафедраси доценти, педагогика фанлари  
номзоди

Ушбу ўқув–услугий мажмуа Гулистон давлат университети Ўқув – методик Кенгаши томонидан (28.08.2017 й. даги, №1- сонли баённома) нашрга тавсия этилган.

## **М У Н Д А Р И Ж А**

1. Маърузалар матни.....	4
2. Амалий машғулотлар ишланмалари.....	165
3. Мустақил таълим мавзулари.....	276
4. Глоссарий.....	277
5. Иловалар	
Фан дастури.....	280
Ишчи фан дастури.....	287
Тестлар.....	297

**“Дастурлаш асослари” фанидан маърузалар  
матни.**

## **1-мавзу: Амалий масалаларни ечиш алгоритмлари**

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

Асосий саволлар:

1. Масала тушунчаси ва унинг қўйилиши.
2. Масалага модел тузиш. Математик моделлаштириш.
3. Алгоритмлаш ва дастурлаш тушунчалари.
4. Дастурни компьютерга киритиш ва унинг натижасини таҳлил қилиш.

**Мавзуга оид таянч тушунча ва иборалар:** Модел, моделнинг турлари, физик модел, биологик модел, абстракт модел, моделлаштириш, математик модел тузиш усуллари, алгоритм, алгоритмнинг берилиш усуллари, алгоритмнинг хоссалари, алгоритмнинг турлари, блок-схема, алгоритмик тил.

Дастурлаш асослари курсини ўрганишда ҳар бир ўқув материали учун мос машқлар танлаш ва улардан дарс жараёнида фойдаланиш алоҳида аҳамиятга эгадир.

Дастурлаш асослари фанида ихтиёрий масалаларни ечиш учун компьютерда масалалар ечиш босқичлари қуйидагича ўрганилади:

- Масаланинг қўйилиши;
- Масалани ечиш моделини тузиш;
- Масалани ечиш усули яъни алгоритмини тузиш;
- Дастур тузиш;
- Дастурни компьютер хотирасига киритиш ва натижа олиш;
- Ечимни асослаш.

Биринчи босқичда масала ва унинг берилишини аниқлаш талаб этилади. Бунинг учун масала нима эканлиги ва унинг турлари ҳақида маълумот бериш керак. Инсон миёсида қабул қилинган маълумотлар қайта ишланиб, уни бирор иш бажаришга ундаса, бу маълумотлар инсон учун масалага айланади. Умуман фан ва тадқиқот жараёнларида бирор муаммони ҳал қилишга қаратилган ибора масала деб қаралади. Масалалар оддий ва мураккаб турларга бўлинади. Информатикада оддий бажариладиган буйруқлар билан ҳал бўладиган масалалар оддий масалалар деб қаралади. Агар масаланинг ҳал бўлиши ундаги шартларнинг бажарилишига боғлиқ бўлса, бундай масала мураккаб масала деб қаралади.

Иккинчи босқичда масалани ечишнинг математик модели тузилади, яъни масалани берилиши ва ечимини математик белгилашлар, маълум қонун-қоидалар асосида математик формулалар яратилади. Яратилган математик формулалар берилган масаланинг модели ҳисобланади.

Учинчи босқичда масалаларни ечиш тартиб қоидаларини умумлаштириш яъни масалани ечишнинг алгоритмини тузиш керак. Бунинг учун алгоритмнинг берилиш усулларини ва турларини билиш лозим. Алгоритмларни сўзлар, жадваллар, графиклар, алгоритмик тил ва схемалар блоки ёрдамида ифодалаш мумкин.

Тўртинчи босқичда масала учун бирор дастурлаш тилида дастур тузиш талаб қилинади. Бунда масаланинг ечиш усулини дастурлаш тилида ифодалаш керак. Дастур тузиш учун энг кўп қўлланиладиган ва юқори имкониятга эга бўлган дастурлаш тилларидан фойдаланиш мақсадга мувофиқ.

Бешинчи босқичда тузилган дастур компьютер хотирасига интерпретатор дастурлари ёрдамида киритилади, таҳрирлаш ишлари амалга оширилади ва дастур ишга туширилиб натижа олинади.

Олтинчи босқичда масала ечими асосланади. Бунда масаланинг алгоритми ва дастури бажарилиши изоҳланади, масаланинг ечими таҳлил қилинади.

Компьютерда масалалар ечиш босқичлари ёрдамида Информатика дарсларида машқлар бажариш ўқувчиларнинг компьютерда ишлаш малакаларини ошириб, ижодий фикрлаш кўникмаларини ривожлантиришга ёрдам беради.

Модел сўзи латинча *modulus* сўзидан олиндиб, ўлчов, меъёр деган маъноларни билдиради. Модел деганда бирор объект ёки объектлар тизимининг образи ёки намунаси тушунилади. Масалан, Ернинг модели деб глобусни, осмон ва ундаги юлдузлар модели деб планетарий экранни, ҳар бир одамнинг модели сифатида эса паспортидаги суратини олиш мумкин.

Модел тузиш жараёни моделлаштириш деб аталади. Моделлаштириш деганда бирор объектни уларнинг моделлари ёрдамида тадқиқ қилиш мавжуд предмет ва ҳодисаларнинг моделларини яшаш ва ўрганиш тушунилади.

Моделлаштириш услубидан ҳозирги замон фанлари кенг фойдаланмоқда. У илмий-тадқиқот жараёнини енгиллаштиради, баъзи ҳолларда эса мураккаб объектларни ўрганишнинг ягона воситасига айланади. Мавҳум объект, олисда жойлашган объектлар, жуда кичик ҳажмдаги объектларни ўрганишда моделлаштиришнинг аҳамияти бекиёсдир. Моделлаштириш услубидан физика, астрономия, биология, иқтисодиёт фанларида объектнинг фақат маълум хусусият ва муносабатларини аниқлашда ҳам фойдаланилади.

Моделларни танлаш воситаларига қараб уларни уч гуруҳга ажратиш мумкин: абстракт, физик ва биологик.

Нарса ёки объектни ҳаёлий тасаввур қилиш орқали формула ва чизмалар ёрдамида ўрганишда қўлланиладиган модел абстракт модел ҳисобланади. Абстракт моделни математик модел деб атаса ҳам бўлади.

Шунинг учун абстракт моделни математик ва математик-мантикий моделларга ажратилади.

Физик моделлар ўрганилаётган объектни кичиклаштириб яшаш ёрдамида тадқиқот ўтказишда қўлланиладиган модел ҳисобланади. Физик моделларга объектларнинг кичиклаштирилган макетлари, турли асбоб ва қурилмалар, тренажёрлар ва бошқалар мисол бўлади. Физик моделлардан самолёт, кема, автомобил, поезд, ГЭС ва бошқа объектларни ўрганиш ёки уларни яратишда қўлланилади.

Биологик модел турли тирик объектлар ва уларнинг қисмлари – молекула, ҳужайра, организм ва бошқаларга хос биологик тузилиш, функция ва жараёнларни моделлаштиришда қўлланилади. Биологик модел одам ва ҳайвонларда учрайдиган маълум бир ҳолат ёки касалликни лабораторияда ҳайвонларда синаб кўриш имконини беради.

### **Назорат саволлари:**

1. Масала тушунчасига таъриф беринг.
2. Компьютерда масалалар ечиш босқичларига тавсиф беринг.
3. Информатикада масалаларнинг қандай турлари мавжуд?
4. Масалага модел тузиш нима учун керак?
5. Алгоритм тузишнинг қандай усуллари мавжуд?
6. Дастур тузишда нималарга эътибор бериш керак?

**2-мавзу:** Алгоритмларни блок-схема кўринишида ифодаланиши

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

### **Асосий саволлар:**

1. Алгоритмларни тасвирлаш усуллари.
2. Схемалар блоки ва уларни тузишда қўлланиладиган геометрик шакллар.
3. Алгоритмларнинг турлари ва уларнинг таснифи.

**Мавзуга оид таянч тушунча ва иборалар:** Математик модел, формула, функция, алгоритм, алгоритмлаш, алгоритмни тасвирлаш усуллари, алгоритмнинг турлари, схемалар блоки, дастур, дастурлаш.

Алгоритмларнинг берилиши ёки уларни тасвирлаш усуллари қуйидагилардан иборат бўлади:

- Алгоритмнинг сўзлар орқали ифодаланиши.
- Алгоритмнинг формулалар ёрдамида берилиши.
- Алгоритмнинг жадвал кўринишида берилиши.
- Алгоритмнинг дастур шаклида ифодаланиши.
- Алгоритмнинг алгоритмик тилда тасвирланиши.
- Алгоритмнинг график (геометрик схемалар) шаклида тасвирланиши.

1. **Алгоритмнинг сўзлар орқали ифодаланиши.** Бу усулда ижрочи учун бериладиган ҳар бир кўрсатма жумлалар, сўзлар орқали буйруқ шаклида берилади.

2. **Алгоритмнинг формулалар билан берилиш** усулидан математика, физика, кимё каби аниқ фанлардаги формулаларни ўрганишда фойдаланилади. Бу усулни баъзан аналитик ифодалаш дейилади.

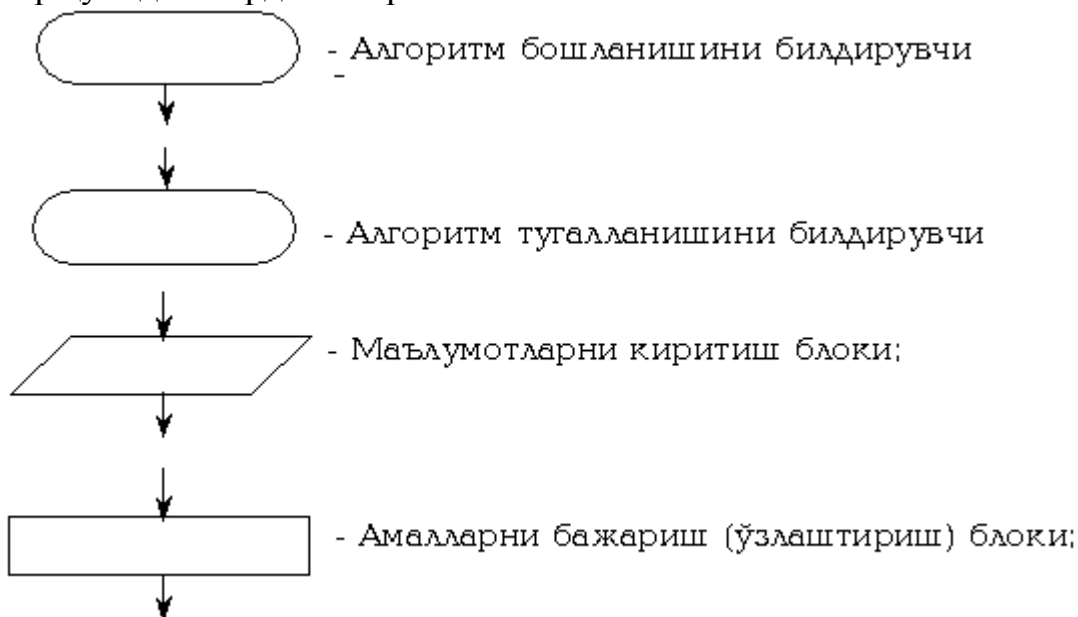
3. **Алгоритмларнинг график шаклида тасвирланишида** алгоритмлар махсус геометрик фигуралар ёрдамида тасвирланади ва бу график кўриниши блок-схема дейилади.

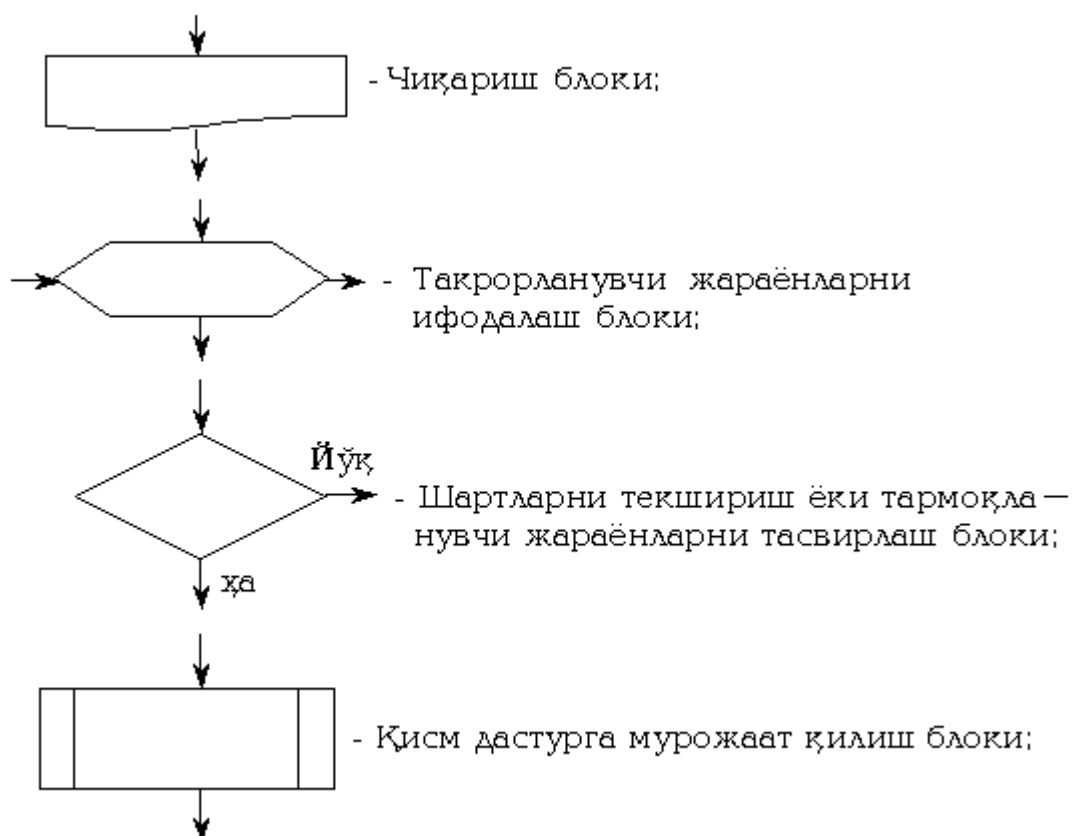
4. **Алгоритмнинг жадвал кўринишида берилиши.** Алгоритмнинг бу тарзда тасвирланишдан ҳам кўп фойдаланамиз. Масалан, мактабда қўлланиб келинаётган тўрт хонали математик жадваллар ёки турли хил лотереялар жадваллари. Функцияларнинг графикларини чизишда ҳам алгоритмларнинг қийматлари жадвали кўринишларидан фойдаланамиз. Бу каби жадваллардан фойдаланиш алгоритмлари содда бўлган туфайли уларни ўзлаштириб олиш осон.

Юқорида кўрилган алгоритмларнинг тасвирлаш усулларининг асосий мақсади, қўйилган масалани ечиш учун зарур бўлган амаллар кетма-кетлигининг энг қулай ҳолатини аниқлаш ва шу билан одам томонидан программа ёзишни янада осонлаштиришдан иборат. Аслида программа ҳам алгоритмнинг бошқа бир кўриниши бўлиб, у инсоннинг компьютер билан мулоқотини қулайроқ амалга ошириш учун мўлжалланган.

Кўпчилик ҳолларда масалани ечиш учун берилган кўрсатмалар ва буйруқлар кетма-кетлигини тасвирлашда алгоритмларнинг график шакллардан яъни схемалар блокидан фойдаланилади.

Блок-схемаларни тузишда фойдаланиладиган асосий содда геометрик фигуралар қуйидагилардан иборат.





Блок-схемалар билан ишлашни яхшилаб ўзлаштириб олиш зарур, чунки бу усул алгоритмларни ифодалашнинг қулай воситаларидан бири бўлиб программа тузишни осонлаштиради, программалаш қобилиятини мустаҳкамлайди. Алгоритмик тилларда блок - схеманинг асосий структураларига махсус операторлар мос келади.

Берилган масаланинг кўринишига қараб, алгоритмларнинг қуйидаги турларидан фойдаланилади:

- Чизиқли алгоритмлар.
- Тармоқланувчи алгоритмлар.
- Такрорланувчи алгоритмлар.

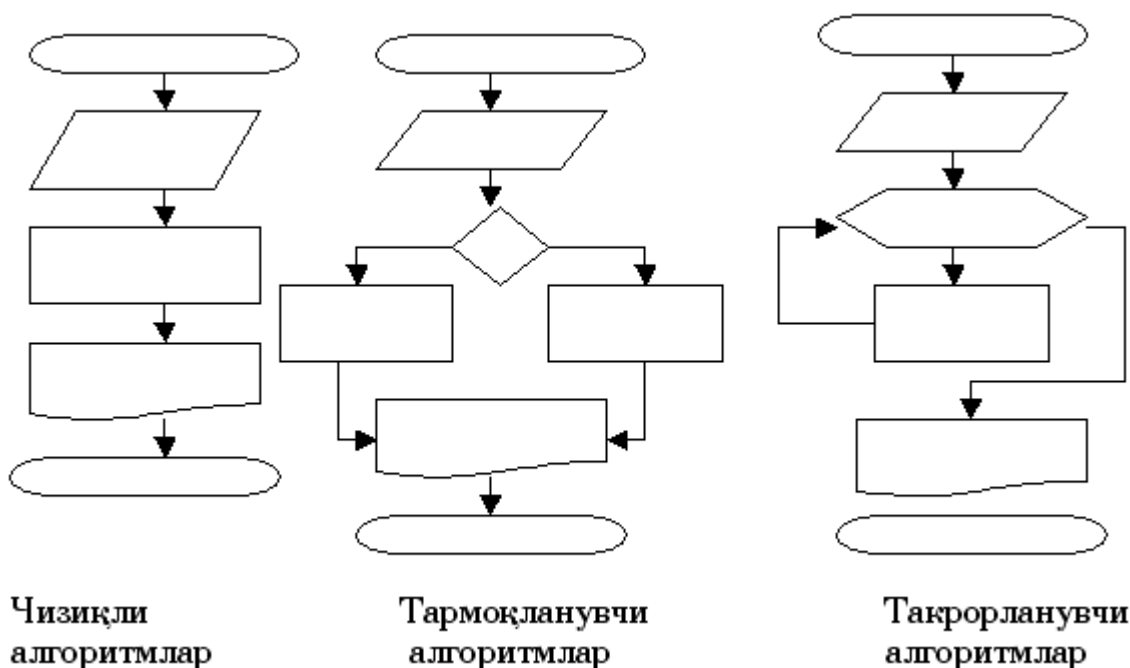
Чизиқли алгоритмлар – оддий кўринишдаги ҳеч қандай шартларга боғлиқ бўлмаган ва тартиб билан фақат кетма-кет бажариладиган жараёнларга тузиладиган алгоритмлардир.

Тармоқланувчи алгоритмлар – бирор шартга мувофиқ бажариладиган кўрсатмалар билан тузиладиган алгоритмлардир.

Такрорланувчи алгоритмлар – кўп марта такрорланадиган қисмни ўз ичига олган ва бирор шарт бажарилгунча давом этадиган алгоритмлардир.

Юқоридаги схемалар блоқи элементларидан фойдаланиб, алгоритм турларининг намунавий кўринишларини қуйидагича ифодалаш мумкин:





Ҳар қандай мураккаб алгоритмни ҳам учта асосий структура ёрдамида тасвирлаш мумкин. Булар кетма-кетлик, айри ва такрорлаш структураларидир. Бу структуралар асосида чизиқли, тармоқланувчи ва такрорланувчи ҳисоблаш жараёнларининг алгоритмларини тузиш мумкин. Умуман олганда алгоритмларни шартли равишда қуйидаги турларга ажратиш мумкин:

- чизиқли алгоритмлар,
- тармоқланувчи алгоритмлар,
- такрорланувчи ёки циклик алгоритмлар,
- ичма-ич жойлашган циклик алгоритмлар,
- рекуррент алгоритмлар,
- такрорланишлар сони олдиндан ноъмалум алгоритмлар,
- кетма-кет яқинлашувчи алгоритмлар.

Ихтиёрий масалага алгоритм тузиш учун алгоритмни ифодалашнинг турли усулларида фойдаланиш мумкин. Алгоритмларни сўзлар ёрдамида ифодалашни кундалик ҳаётимизда кўп учрайдиган «Чой дамлаш алгоритми» мисолида қуйидагича тасвирлаш мумкин:

- \* Чойнак қопқоғи очилсин;
- \* Чойнакни қайноқ сув билан чайқалсин;
- \* Чойнакка бир чой қошиқ куруқ чой солинсин;
- \* Чойнакка тўлгунча қайнаган сув қуйилсин;
- \* Чойнак қопқоғи ёпилсин;
- \* Чойнакни сочик билан ўраб, беш дақиқа дам берилсин.

Алгоритмни алгоритмик тил ёрдамида тузиш масаласини «Квадрат тенгламани ечиш» алгоритми орқали қуйидагича келтириш мумкин:

алг Квадрат тенгламани ечиш (ҳақ а,в,с,  $x_1, x_2$ , лит у)  
арг а,в,с

натижа  $x_1, x_2, y$   
бошл ҳақ  $D$   
 $D := b^2 - 4 * a * c$   
агар  $D < 0$   
     бўлса у:қ «ечим йўқ»  
     акс ҳолда у:қ «ечими бор»  
         агар  $D = 0$   
             бўлса  $x_1 := -\frac{b}{2a}$   
                  $x_2 := x_1$   
             акс ҳолда  $x_1 := \frac{-b + \sqrt{D}}{2a}$   
                  $x_2 := \frac{-b - \sqrt{D}}{2a}$   
         ҳал бўлди  
     ҳал бўлди  
там

Назорат саволлари:

1. Масалани ЭХМда ечиш бочқичлари нечта?
2. Алгоритм ва унинг хоссаларини тушунтириб беринг.
3. Блок-схемада иштирок этувчи блокларни санаб беринг.
4. Алгоритм тушунчасига таъриф беринг.
5. Алгоритм тузишда нималарга эътибор бериш керак?
6. Алгоритмнинг қандай турлари бор?
7. Алгоритмнинг қандай берилиш усуллари мавжуд?
8. Алгоритмнинг берилиш усулларига мисол келтиринг?
9. Алгоритмларнинг қандай хоссалари бор?
10. Алгоритмлашнинг қандай аҳамияти бор?
11. Герон формуласи бўйича учбурчак юзасини ҳисоблаш алгоритми тузилсин.
12.  $z = x^2 + 2x^4 + b$  функцияни  $x = 3$  қийматида ҳисоблаш алгоритмининг блок схемаси тузилсин.

**3-мавзу:** C++ тили синтаксиси ва унинг лексик асоси..

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

1. C++ тили тарихи
2. C++ тили алфавити ва лексемалар
3. Идентификаторлар ва калит сўзлар

Мавзудаги асосий таянч тушунчалар ва иборалар: Дастурлаш тили, юқори ва қуйи даражадаги дастурлаш тили, лексема, идентификатор, калит сўз, функция, компилятор.

Кейинги йилларда амалий дастурчиларга жуда кўп интеграцион дастур тузиш муҳитлари таклиф этилаяпти. Бу муҳитлар у ёки бу имкониятлари билан бир-биридан фарқ қилади. Аксарият дастурлаштириш муҳитларининг фундаментал асоси С++ тилига бориб тақалади. Биз ушбу мавзуда қуйидаги саволларга жавоб оламиз:

- Нима учун С++ тили дастурий маҳсулотларни ишлаб чиқиш соҳасида стандарт бўлиб қолди?
- С++ тилида дастур ишлаб чиқишни қандай ўзига хос томонлари бор?
- С++ тилида дастур қандай ёзилади ва компиляция қилинади?

### **С++ тили тарихи**

Биринчи электрон ҳисоблаш машиналари пайдо бўлиши билан дастурлаш тиллари эволюцияси бошланади. Дастлабки компьютерлар иккинчи жаҳон уруши вақтида артиллерия снарядларининг ҳаракат траекториясини ҳисоб-китоб қилиш мақсадида қурилган эди. Олдин дастурчилар энг содда машина тилини ўзида ифодаловчи компьютер командалари билан ишлаганлар. Бу командалар нол ва бирлардан ташкил топган узун қаторлардан иборат бўлар эди. Кейинчалик, инсонлар учун тушунарли бўлган машина командаларини ўзида сақловчи (масалан, ADD ва MOV командалари) ассемблер тили яратилди. Шу вақтларда BASIC ва COBOL сингари юқори сатҳли тиллар ҳам пайдо бўлдики, бу тиллар туфайли сўз ва гапларнинг мантиқий конструкциясидан фойдаланиб дастурлаш имконияти яратилди. Бу командаларни машина тилига интерпретаторлар ва компиляторлар кўчиради эди. Интерпретатор дастурни ўқиш жараёнида унинг командаларини кетма - кет машина тилига ўтказди. Компилятор эса яхлит программа кодини бирор бир оралик форма - объект файлига ўтказди. Бу босқич компиляция босқичи дейилади. Бундан сўнг компилятор объектли файлни бажарилувчи файлга айлантирадиган компановка дастурини чақиради.

Интерпретаторлар билан ишлаш осонроқ, чунки дастур командалари қандай кетма - кетликда ёзилган бўлса шу тарзда бажарилади. Бу эса дастур бажарилишини назорат қилишни осонлаштиради. Компилятор эса компиляция ва компановка каби қўшимча босқичлардан иборат бўлганлиги учун улардан ҳосил бўладиган бажарилувчи файлни таҳлил қилиш ва ўзгартириш имконияти мавжуд эмас. Фақатгина компиляция қилинган файл тезроқ бажарилади, чунки бундаги командалар компиляция жараёнида машина тилига ўтказилган бўлади.

C++ каби компиляция қилувчи дастурлаш тилларини яна бир афзаллиги ҳосил бўлган дастур компьютерда компиляторсиз ҳам бажарилаверади. Интерпретация қилувчи тилларда эса тайёр дастурни ишлатиш учун албатта мос интерпретатор дастури талаб қилинади.

Айрим тилларда (масалан, VISUAL BASIC) интерпретатор ролини динамик библиотекалар бажаради. Java тилининг интерпретатори эса виртуал машинадир (Virtual Machine, ёки VM). Виртуал машиналар сифатида одатда броузер (Internet Explorer ёки Netscape) лар қўлланилади.

Кўп йиллар давомида дастурларнинг асосий имконияти унинг қисқалиги ва тез бажарилиши билан белгиланиб келинар эди. Дастурни кичикроқ қилишга интилиш компьютер хотирасини жуда қимматлиги билан боғлиқ бўлса, унинг тез бажарилишига қизиқиш процессор вақтининг қимматбаҳолигига боғлиқ эди. Лекин компьютерларнинг нархи тушиши билан дастур имкониятини баҳолаш мезони ўзгарди. Ҳозирги кунда дастурчининг иш вақти бизнесда ишлатиладиган кўпгина компьютерларнинг нархидан юқори. Ҳозирда профессионал тарзда ёзилган ва осон эксплуатация қилинадиган дастурларга талаб ошиб бормокда. Эксплуатациянинг оддийлиги, конкрет масалани ечиш билан боғлиқ бўлган талабни озроқ ўзгаришига, дастурни ортиқча чиқимларсиз осон мослаштириш билан изоҳланади.

### **C++ тили алфавити ва лексемалар**

C++ тили алфавити ва лексемаларига қуйидагилар киради:

- катта ва кичик латин алфавити ҳарфлари;
- рақамлар - 0,1,2,3,4,5,6,7,8,9;
- махсус белгилар:“ { } | [ ] ( ) + - / % \ ; ‘ : ? < = > \_ ! & ~ # ^ . \*

Алфавит белгиларидан тилнинг лексемалари шакллантирилади: идентификаторлар; калит (хизматчи ёки захиранланган) сўзлар; ўзгармаслар; амаллар белгиланишлари; ажратувчилар.

### **Идентификаторлар ва калит сўзлар**

Программалаш тилининг муҳим таянч тушунчаларидан бири - идентификатор тушунчасидир. *Идентификатор* - бу катта ва кичик латин ҳарфлари, рақамлар ва таг чизик (‘\_’) белгиларидан ташкил топган ва рақамдан бошланмайдиган белгилар кетма-кетлигига айтилади. Идентификаторларда ҳарфларнинг регистрлари (катта ёки кичиклиги) ҳисобга олинади. Масалан, *RUN*, *run*, *Run* - бу ҳар хил идентификаторлардир.

Идентификаторлар калит сўзлар, ўзгарувчилар, функциялар, нишонлар ва бошқа объектларни номлашда ишлатилади.

C++ тилининг калит сўзларига қуйидагилар киради:

*asm, auto, break, case, catch, char, class, const, continue, default, delete, do, double, else, enum, explicit, extern, float, for, friend, goto, if, inline, int, long, mutable, new, operator, private, protected, public, register, return, short,*

*signed, sizeof, static, struct, swith, template, this, throw, try, typedef, typename, union, unsigned, virtual, void, volatile, while.*

Юқорида келтирилган идентификаторларни бошқа мақсадда ишлатиш мумкин эмас.

Процессор регистрларини белгилаш учун қуйидаги сўзлар ишлатилади:

*\_AH, \_AL, \_AX, \_EAX, \_BH, \_BL, \_BX, \_EBX, \_CL, \_CH, \_CX, \_ECX, \_DH, \_DL, \_DX, \_EDX, \_CS, \_ESP, \_EBP, \_FS, \_GS, \_DI, \_EDI, \_SI, \_ESI, \_BP, \_SP, \_DS, \_ES, \_SS, \_FLAGS.*

Булардан ташқари «\_\_» (иккита тагчизик) белгиларидан бошланган идентификаторлар кутубхоналар учун захиранланган. Шу сабабли '\_' ва «\_\_» белгиларни идентификаторнинг биринчи белгиси сифатида ишлатмаган маъкул. Идентификатор белгилар орасида пробел ишлатиш мумкин эмас, зарур бўлганда унинг ўрнига '\_' ишлатиш мумкин: *Cilindr\_radiusi, ailana\_diametiri.*

## НАЗОРАТ ТОПШИРИҚЛАРИ:

1. C++ тили алфавити.
2. C++ тилидаги калит сўзлар.
3. Дастурлаш тиллари классификацияси.

### 4-мавзу: C++ тили дастурининг тузилиши ва шакли.

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

### ДАРС Р Е Ж АСИ (асосий саволлар):

1. C++ тилидаги программа тузилиши ва унинг компиляцияси
2. C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

C++ тилидаги программа тузилиши ва унинг компиляцияси  
C++ тилида программа тузилиши тушунтириш учун содда программа келтирамиз.

```
#include <iostream.h> // сарлавжа файлини қўшиш
int main()             // бош функция тавсифи
{                       // блок бошланиши
```

```

cout<<"Salom Olam!\n"; // сатрни чоп этиш
return 0;           // функция қайтарадиган қиймат
}                  // блок тугаши

```

Программанинг 1-сатрида “*#include*” препроцессор кўрсатмаси бўлиб, программа кодига стандарт оқимли ўқиш-ёзиш функциялари ва унинг ўзгарувчилари эълони жойлашган «*iostream.h*» сарлавҳа файлини кўшади (мнемоника: ‘*i*’(*input*) - киритиш (ўқиш); ‘*o*’(*output*) - чиқариш (ёзиш); “*stream*”- оқим; ‘*h*’(*head*) –сарлавҳа). Келишув бўйича стандарт оқим экранга чиқариш ҳисобланади. Кейинги қаторларда программанинг ягона, асосий функцияси - *main()* функцияси тавсифи келтирилган. Шунинг қайд этиш керакки, C++ программасида албатта *main()* функцияси бўлиши шарт ва программа шу функцияни бажариш билан ўз ишини бошлайди. Функция номи олдидаги “*int*” калит сўзи функция бажарилиши натижаси у бутун сон қийматини қайтариши кераклигини билдиради. Бу ҳолат функция-нинг математикадаги тавсифига мос келади. Кейинги қатордан функция танаси - фигурали кавсга олинган амаллар кетма-кетлиги келади. Бизнинг ҳолда функция танаси иккита амалдан иборат. Биринчиси, консол режимида белгилар кетма-кетлигини оқимга чиқариш амали қўлланилган. Бунинг учун «*iostream.h*» сарлавҳа файлида аниқланган *cout* объектидан фойдаланилган. Унинг формати қуйидаги кўринишда:

```
cout << <ифода>;
```

Бу ерда “<<” – маълумот узатиш амали («..га жойлаштир»), <ифода> сифатида ўзгарувчи ёки синтаксиси тўғри ёзилган ва қандайдир қиймат қабул қилувчи тил ифодаси келиши мумкин (*кейинчалик, бурчак қавс ичига олинган ўзбекча сатр остини тил таркибига кирмайдиган тушунча деб қабул қилиш керак*).

Иккинчиси, функция ўз ишини тугатганлигини англатувчи ва ундан чиқишни амалга оширувчи “*return 0;*” операторидир. Одатда, бажарилиши нормал тугаган функциялар операцион системага 0 қийматини қайтаради. Шу қоидага риоя қилган ҳолда программа ҳам 0 қийматини қайтаради.

Бажарилувчи программани ҳосил қилиш учун програма матни компиляция қилиниши керак. Компиляция жараёнининг ўзи ҳам иккита босқичдан ташкил топади. Бошида препроцессор ишлайди, у матндаги компиляция директиваларини бажаради, хусусан *#include* директиваси бўйича кўрсатилган кутубхоналардан C++ тилида ёзилган модулларни программа таркибига киритади. Шундан сўнг кенгайтирилган программа матни компиляторга узатилади. Компиля-тор ўзи ҳам программа бўлиб, унинг учун кирувчи маълумот бўлиб, C++ тилида ёзилган программа матни ҳисобланади. Компилятор программа матнини лексема (атомар) элементларга ажратади ва уни лексик, кейинчалик синтаксик таҳлил қилади. Лексик таҳлил жараёнида у матнни лексемаларга ажратиш учун «пробел ажратувчи-сини» ишлатади. Пробел ажратувчисига - пробел

белгиси (' $\backslash$ '), ' $\backslash t$ ' - табуляция белгиси, ' $\backslash n$ ' - кейинги қаторга ўтиш белгиси, бошқа ажратувчилар ва изоҳлар киради.

C++ тилида бажарилувчи программа яратиш босқичлар:

1. Матн таҳририда (одатда программалаш муҳитининг таҳририда) программа матни терилади, бу файлнинг кенгаймаси ".cpp" бўлади, масалан "salom.cpp";

2. Программа матни ёзилган файл компиляторга узатилади. Агар хатоликлар бўлса, улар тўғриланади;

3. Компилятор томонидан программа матнига сарлавҳа файл-лар киритилади ("include " препроцессор кўрсатмасига мувофиқ);

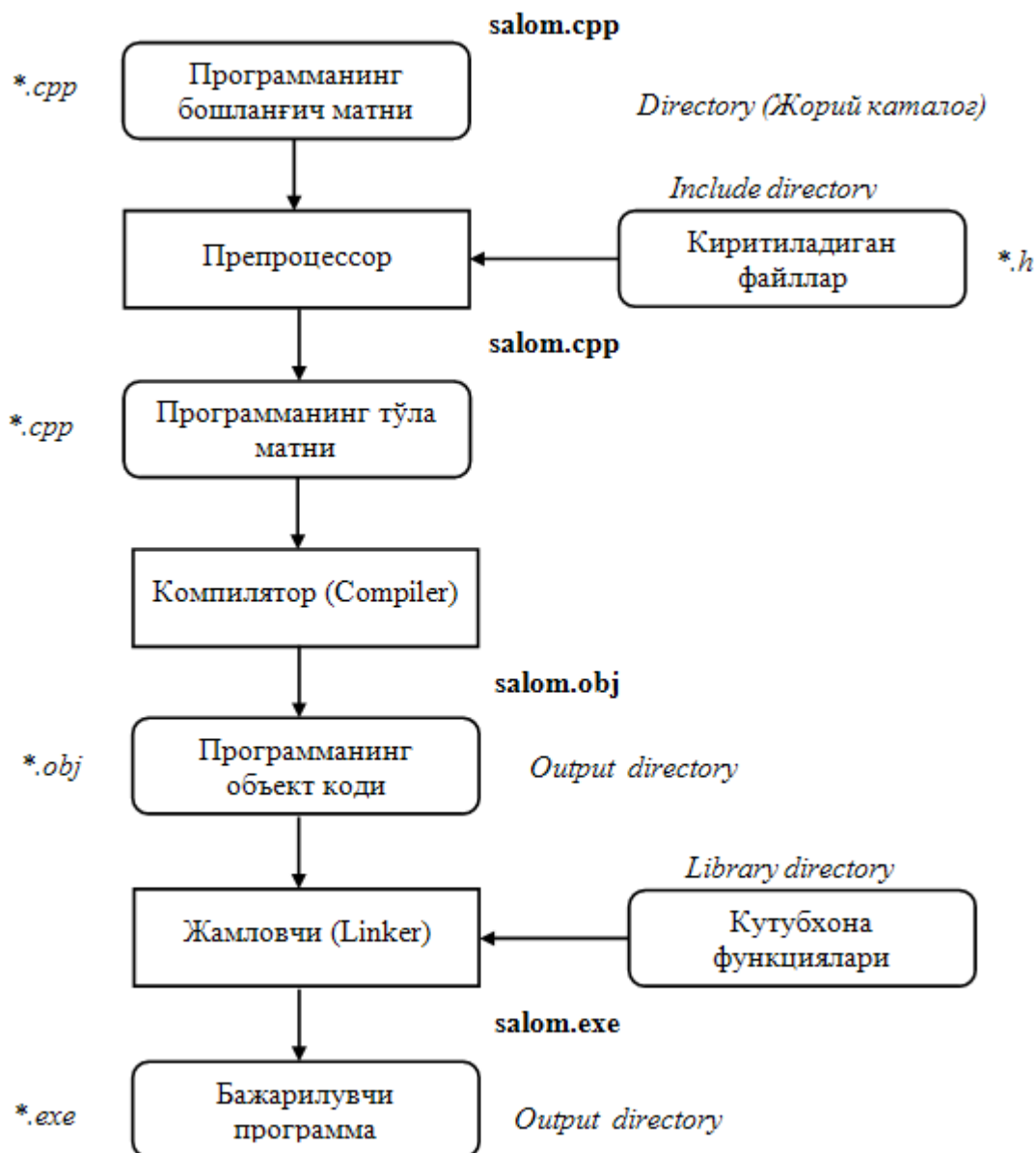
4. Компилятор ".obj" ("salom.obj") кенгайтмали объект файли ҳосил қилади;

5. Компоновка (жамловчи) ёрдамида объект файлга кутубхоналардан зарур функциялари қўшилади ва ".exe" кенгаймали бажари-лувчи файл - программа ҳосил бўлади ("salom.exe");

6. Программани ишга тушириш учун буйруқ сатрида программа номини териш ва "Enter" тугмасини босиш етарли.

Босқичларда юзага келувчи файлларнинг номлари бошланғич матн файлининг номи билан бир хил бўлади (1.1-расм).

Юқорида келтирилган программа бажарилиши натижасида экранга "Salom Olam!" сатри чоп этилади.



1.1-расм. Бажарилувчи программани тайёрлаш босқичлари

Программа матни тушунарли бўлиши учун изоҳлар ишлатилади. Изоҳлар компилятор томонидан «ўтказиб» юборилади ва улар программа амал қилишига ҳеч қандай таъсир қилмайди.

C++ тилида изоҳлар икки кўринишда ёзилиши мумкин.

Биринчисида “/\*” дан бошланиб, “\*/” белгилар оралиғида жойлашган барча белгилар кетма-кетлиги изоҳ ҳисобланади, иккинчиси «сатрий изоҳ» деб номланади ва у “//” белгилардан бошланган ва сатр охиригача ёзилган белгилар кетма-кетлиги бўлади. Изоҳнинг биринчи кўринишида ёзилган изоҳлар бир неча сатр бўлиши ва улардан кейин C++ операторлари давом этиши мумкин.

Мисол.

```

int main()
{

```



```

// бу қатор изоҳ ҳисобланади
int a=0; //int d;
int c;
/* int b=15 */
/* - изоҳ бошланиши
a=c;
изоҳ тугаши */
return 0;
}

```

Программада  $d$ ,  $b$  ўзгарувчилар эълонлари инобатга олинмайди ва  $a=c$  амали бажарилмайди.

Масалан:

```

int uzg=324;
cout<<uzg; // бутун сон чоп этилади

```

Берилганларни стандарт оқимдан (одатда клавиатурадан) ўқиш куйидаги форматда амалга оширилади:

```
cin >> <ўзгарувчи>;
```

Бу ерда <ўзгарувчи> оқимдан қиймат қабул қилувчи ўзгарувчининг номи.

Мисол:

```

int Yosh;
cout<<"Yoshingizni kiriting_" ;
cin>>Yosh;

```

Бутун турдаги *Yosh* ўзгарувчиси киритилган қийматни ўзлаш-тиради. Киритилган қийматни ўзгарувчи турига мос келишини текши-риш масъулияти программа тузувчисининг зиммасига юкланади.

Бир пайтнинг ўзида пробел воситасида бир нечта ва ҳар хил турдаги қийматларни оқимдан киритиш мумкин. Қиймат киритиш “Enter” тугмасини босиш билан тугайди. Агар киритилган қийматлар сони ўзгарувчилар сонидан кўп бўлса, «ортиқча» қийматлар буфер хотирада сақланиб қолади.

```

#include <iostream.h>
int main()
{
    int x,y;
    float z;
    cin>>x>>y>>z;
    cout<<"O'qilgan qiymatlar\n";
    cout<<x<<' \t' <<y<<' \t' <<z;
    return 0;
}

```

Ўзгарувчиларга қиймат киритиш учун клавиатура орқали

## 10 20 3.14 <enter>

ҳаракати амалга оширилади. Шунинг қайд этиш керакки, оқимга қиймат киритишда пробел ажратувчи ҳисобланади. Ҳақиқий соннинг бутун ва каср қисмлари ‘.’ белгиси билан ажратилади.

### НАЗОРАТ ТОПШИРИҚЛАРИ:

1. C++ тилида дастур структураси.
2. Бажарилувчи файл ҳосил қилиш босқичлари.
3. Компиляция ва компоновка фарқлари

**5-мавзу:** Берилганлар турлари. C++ тилининг таянч турлари.

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

### ДАРС Р Е Ж А С И (асосий саволлар):

1. Ўзгармаслар
2. Берилганлар турлари ва ўзгарувчилар
3. C++ тилининг таянч турлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Ўзгармас, ўзгарувчи, ҳақиқий, бутун, белгили, мантикий.

### Ўзгармаслар

*Ўзгармас (литерал)* - бу фиксирланган сонни, сатрни ва белгини ифодаловчи лексемадир.

Ўзгармаслар бешта гуруҳга бўлинади - *бутун, ҳақиқий (сузувчи нуқтали), санаб ўтилувчи, белги (литерли) ва сатр («стринг», литерли сатр).*

Компилятор ўзгармасни лексема сифатида аниқлайди, унга хотирадан жой ажратади, кўриниши ва қийматига (турига) қараб мос гуруҳларга бўлади.

**Бутун ўзгармаслар.** Бутун ўзгармаслар қуйидаги форматларда бўлади:

- ўнлик сон;
- саккизлик сон;
- ўн олтилик сон.

*Ўнлик ўзгармас* 0 рақамидан фарқли рақамдан бошланувчи рақамлар кетма-кетлиги ва 0 ҳисобланади: **0 ; 123 ; 7987 ; 11 .**

*Манфий ўзгармас* - бу ишорасиз ўзгармас бўлиб, унга фақат ишорани ўзгартириш амали қўлланилган деб ҳисобланади.

*Саккизлик ўзгармас* 0 рақамидан бошланувчи саккизлик санок системаси (0,1,...,7) рақамларидан ташкил топган рақамлар кетма-кетлиги:

**023; 0777; 0.**

*Ўн олтилик ўзгармас* “0x” ёки “0X” белгиларидан бошланадиган ўн олтилик санок системаси рақамларидан иборат кетма-кетлик ҳисобланади. Ўн олтилик сон ёзилишидаги ҳарф белгилари ихтиёрий регистрларда берилиши мумкин:

**0x1A; 0X9F2d; 0x23aAbC.**

Компилятор соннинг қийматига қараб унга мос турни белгилайди. Агар тилда белгиланган турлар программа тузувчини қаноатлантирмаса, у ошкор равишда турни кўрсатиши мумкин. Бунинг учун бутун ўзгармас рақамлари охирига, пробелсиз *l* ёки *L (long)*, *u* ёки *U (unsigned)* ёзилади. Зарур ҳолларда битта ўзгармас учун бу белгиларнинг иккитасини ҳам ишлатиш мумкин:

**451u; 012U1; 0xA2L.**

**Ҳақиқий ўзгармаслар.** *Ҳақиқий ўзгармаслар* - сузувчи нуқтали сон бўлиб, у икки хил форматда берилиши мумкин:

- *ўнлик фиксирланган нуқтали* форматда. Бу кўринишда сон нуқта орқали ажратилган бутун ва каср қисмлар кўринишида бўлади. Соннинг бутун ёки каср қисми бўлмаслиги мумкин, лекин нуқта албатта бўлиши керак. Фиксирланган нуқтали ўзгармасларга мисоллар: **24.56; 13.0; 66.; .87;**

- *экспоненциал шаклда ҳақиқий ўзгармас* 6 қисмдан иборат бўлади:

- 1) бутун қисми (ўнли бутун сон);
- 2) ўнли каср нуқта белгиси;
- 3) каср қисми (ўнлик ишорасиз ўзгармас);
- 4) экспонента белгиси ‘e’ ёки ‘E’;
- 5) ўн даражаси кўрсаткичи (ўнли бутун сон);
- 6) қўшимча белгиси ( ‘F’ ёки ‘f’ , ‘L’ ёки ‘l’ ).

Экспоненциал шаклдаги ўзгармас сонларга мисоллар: **1e2; 5e+3; .25e4; 31.4e-1 .**

**Белги ўзгармаслар.** *Белги ўзгармаслар* - апострофлар (‘’- белгилари) ичига олинган алоҳида белгилардан ташкил топади ва у *char* калит сўзи билан аниқланади. Белги ўзгармас учун хотирада бир байт жой ажратилади ва унда бутун сон кўринишидаги белгининг ASCII коди жойлашади. Қуйидагилар белги ўзгармасларга мисол бўлади: ‘e’ , ‘@’ , ‘7’ , ‘z’ , ‘W’ , ‘+’ , ‘ш’ , ‘\*’ , ‘a’ , ‘s’ .

1.1-жадвал. C++ тилида escape -белгилар жадвали

Escape	Ички код	Номи	Амал
--------	----------	------	------

Белгилари	(16 сон)		
\\	0x5C	\	Тескари ён чизиқни чоп этиш
\'	0x27	'	Апострофни чоп этиш
\"	0x22	"	Қўштирноқни чоп этиш
\?	0x3F	?	Сўроқ белгиси
\a	0x07	bel	Товуш сигналини бериш
\b	0x08	bs	Курсорни 1 белги ўрнига орқага қайтариш
\f	0x0C	ff	Саҳифани ўтказиш
\n	0x0A	lf	Қаторни ўтказиш
\r	0x0D	cr	Курсорни айна қаторнинг бошига қайтариш
\t	0x09	ht	Навбатдаги табуляция жойига ўтиш
\v	0x0D	vt	Вертикал табуляция (пастга)
\000	000		Саккизлик коди
\xNN	0xNN		Белги ўн олтилик коди билан берилган

Айрим белги ўзгармаслар '\ ' белгисидан бошланади, бу белги биринчидан, график кўринишга эга бўлмаган ўзгармасларни белги-лайди, иккинчидан, махсус вазифалар юкланган белгилар - апостроф белгиси, савол белгисини ('?'), тескари ён чизиқ белгисини ('\') ва иккита қўштирноқ белгисини (' ' ') чоп қилиш учун ишлатилади. Ундан ташқари, '\ ' белгиси ёрдамида белгини, унинг ASCII кодини саккизлик ёки ўн олтилик сон кўринишида ёзиш орқали бериш мумкин. Бундай белгидан бошланган белгилар *escape* кетма-кетликлар дейилади (1.1-жадвал).

C++ тилида қўшимча равишда *wide* ҳарфли ўзгармаслар ва кўп белгили ўзгармаслар аниқланган.

*wide* ҳарфли ўзгармаслар тури миллий кодларни белгилаш учун киритилган бўлиб, у *wchar\_t* калит сўзи билан берилади, ҳамда хотирада 2 байт жой эгаллайди. Бу ўзгармас *L* белгисидан бошланади:

**L' \013\022' , L' cc'**

Кўп белгили ўзгармас тури *int* бўлиб, у тўртта белгидан иборат бўлиши мумкин:

**' abc' , '\001\002\003\004' .**

**Сатр ўзгармаслар.** Иккита қўштирноқ ичига олинган белгилар кетма-кетлиги *satr* ўзгармас дейилади:

**"Bu satr o'zgarmas va uning nomi string\n"**

Сатр ичида *escape* кетма-кетлиги ҳам ишлатилиши мумкин, фақат бу кетма-кетлик апострофсиз ёзилади.

Пробел билан ажратиб ёзилган сатрлар компилятор томонидан ягона сатрга уланади (конкантенация):

**"Satr - bu belgilar massivi" /\* бу сатр кейинги сатрга қўшилади \*/ , uning turi char[]";**

Бу ёзув

```
"Satr - bu belgilar massivi, uning turi char[]";
```

ёзуви билан эквивалент ҳисобланади.

Узун сатрни бир нечта қаторга ёзиш мумкин ва бунинг учун қатор охирида ‘\’ белгиси қўйилади:

```
"Kompilyator har bir satr uchun kompyuter  
xotirasida\  
satr uzunligiga teng sondagi baytlardagi alohida  
\  
xotira ajratadi va bitta - 0 qiymatli bayt  
qo'shadi";
```

Юқоридаги учта қаторда ёзилган сатр келтирилган. Тескари ён чизик (‘\’) белгиси кейинги қаторда ёзилган белгилар кетма-кетлигини юқоридаги сатрга қўшиш кераклигини билдиради. Агар қўшиладиган сатр бошланишида пробеллар бўлса, улар ҳам сатр таркибига киради.

Сатр хотирада жойлашганда унинг охирига ‘\0’ (0 кодли белги) қўйилади ва бу белги сатр тугаганлигини билдиради. Шу сабабли сатр узунлиги, унинг «ҳақиқий» қийматидан биттага кўп бўлади.

### **Берилганлар турлари ва ўзгарувчилар**

Программа бажарилиши пайтида қандайдир берилганларни сақлаб туриш учун ўзгарувчилар ва ўз

гармаслардан фойдаланилади. *Ўзгарувчи* - программа объекти бўлиб, хотирадаги бир нечта ячейка-ларни эгаллайди ва берилганларни сақлаш учун хизмат қилади. Ўзгарувчи номга, ўлчамга ва бошқа атрибутларга - кўриниш соҳаси, амал қилиш вақти ва бошқа хусусиятларга эга бўлади. Ўзгарувчи-ларни ишлатиш учун улар албатта эълон қилиниши керак. Эълон натижасида ўзгарувчи учун хотирадан қандайдир соҳа захираланади, соҳа ўлчами эса ўзгарувчининг конкрет турига боғлиқ бўлади. Шуни қайд этиш зарурки, битта турга ҳар хил аппарат платформаларда турлича жой ажратилиши мумкин.

Ўзгарувчи эълони унинг турини аниқловчи калит сўзи билан бошланади ва ‘=’ белгиси орқали бошланғич қиймат берилади (ҳар доим шарт эмас). Битта калит сўз билан бир нечта ўзгарувчиларни эълон қилиш мумкин. Бунинг учун ўзгарувчилар бир-биридан ‘,’ белгиси билан ажратилади. Эълонлар ‘;’ белгиси билан тугайди. Ўзгарувчи номи 255 белгидан ошмаслиги керак.

### **C++ тилининг таянч турлари**

C++ тилининг таянч турлари, уларнинг байтлардаги ўлчамлари ва қийматларининг чегаралари 1.2-жадвалда келтирилган.

**Бутун сон турлари.** Бутун сон қийматларни қабул қиладиган ўзгарувчилар *int* (бутун), *short* (қисқа) ва *long* (узун) калит сўзлар билан

аниқланади. Ўзгарувчи қийматлари ишорали бўлиши ёки *unsigned* калит сўзи билан ишорасиз сон сифатида қаралиши мумкин (1-иловага қаранг).

**Белги тури.** Белги туридаги ўзгарувчилар *char* калит сўзи билан берилади ва улар ўзида белгининг ASCII кодини сақлайди. Белги туридаги қийматлар нисбатан мураккаб бўлган тузилмалар - сатрлар, белгилар массивлари ва ҳакозаларни ҳосил қилишда ишлатилади (2-иловага қаранг).

1.2-жадвал. C++ тилининг таянч турлари

Тур номи	Байтлардаги ўлчам	Қиймат чегараси
Bool	1	<i>true</i> ёки <i>false</i>
unsigned short int	2	0..65535
short int	2	-32768..32767
unsigned long int	4	0..42949667295
long int	4	-2147483648..2147483647
int (16 разрядли)	2	-32768..32767
int (32 разрядли)	4	-2147483648..2147483647
unsigned int (16 разрядли)	2	0..65535
unsigned int (32 разрядли)	4	0..42949667295
unsigned char	1	0..255
Char	1	-128..127
Float	4	1.2E-38..3.4E38
Double	8	2.2E-308..1.8E308
long double (32 разрядли)	10	3.4e-4932..-3.4e4932
Void	2 ёки 4	-

**Ҳақиқий сон тури.** Ҳақиқий сонлар *float* калит сўзи билан эълон қилинади. Бу турдаги ўзгарувчи учун хотирада 4 байт жой ажратилади ва <ишора><тартиб><мантисса> қолипида сонни сақлай-ди (1-иловага қаранг). Агар касрли сон жуда катта (кичик) қиймат-ларни қабул қиладиган бўлса, у хотирада 8 ёки 10 байтда иккиланган аниқлик кўринишида сақланади ва мос равишда *double* ва *long double* калит сўзлари билан эълон қилинади. Охирги ҳолат 32-разрядли платформалар учун ўринли.

**Мантикий тур.** Бу турдаги ўзгарувчи *bool* калит сўзи билан эълон қилинади. У турдаги ўзгарувчи 1 байт жой эгаллайди ва 0 (*false*, ёлғон) ёки 0 қийматидан фарқли қиймат (*true*, рост) қабул қиладди. Мантикий турдаги ўзгарувчилар қийматлар ўртасидаги муносабат-ларни ифодаладиган мулоҳазаларни рост ёки ёлғон эканлигини тавсифлашда қўлланилади ва улар қабул қиладиган қийматлар математик мантиқ қонуниятларига асосланади.

*Математик мантиқ* - фикрлашнинг шакли ва қонуниятлари ҳақидаги фан. Унинг асосини мулоҳазалар ҳисоби ташкил қиладди. *Мулоҳаза* - бу ихтиёрий жумла бўлиб, унга нисбатан рост ёки ёлғон

фикрни билдириш мумкин. Масалан «3>2», «5 - жуфт сон», «Москва-Украина пойтахти» ва ҳакозо. Лекин «0.000001 кичик сон» жумласи мулоҳаза ҳисобланмайди, чунки «кичик сон» тушунчаси жуда ҳам нисбий, яъни кичик сон деганда қандай сонни тушуниш кераклиги аниқ эмас. Шунинг учун юқоридаги жумлани рост еки ёлғонлиги ҳақида фикр билдириш қийин.

Мулоҳазалар ростлиги ҳисоблаш жараёнида юзага келадиган ҳолатларга боғлиқ равишда ўзгариши мумкин. Масалан «бугун - чоршанба» жумласини рост ёки ёлғонлиги айна қаралаётган кунга боғлиқ. Худди шундай « $x < 0$ » жумласи  $x$  ўзгарувчисининг айна пайтдаги қийматига мос равишда рост ёки ёлғон бўлади.

C++ тилида мантиқий тур номи англиялик математик Жорж Бул шарафига *bool* сўзи билан ифодаланган. Мантиқий амаллар «Бул алгебраси» дейилади.

Мантиқий мулоҳазалар устида учта амал аниқланган:

1) *инкор* -  $A$  мулоҳазани инкори деганда  $A$  рост бўлганда ёлғон ва ёлғон бўлганда рост қиймат қабул қилувчи мулоҳазага айтилади. C++ тилида инкор - «!» белгиси билан берилади. Масалан,  $A$  мулоҳаза инкори «!A» кўринишида ёзилади;

2) *конъюнкция* - иккита  $A$  ва  $B$  мулоҳазалар конъюнкцияси ёки мантиқий кўпайтмаси « $A \& \& B$ » кўринишга эга. Бу мулоҳаза фақат  $A$  ва  $B$  мулоҳазалар рост бўлгандагина рост, акс ҳолда ёлғон бўлади (одатда « $\& \&$ » амали «ва» деб ўқилади). Масалан «бугун ойнинг 5 куни ва бугун чоршанба» мулоҳазаси ойнинг 5 куни чоршанба бўлган кунлар учунгина рост бўлади;

3) *дизъюнкция* - иккита  $A$  ва  $B$  мулоҳазалар дизъюнкцияси ёки мантиқий йиғиндиси « $A \parallel B$ » кўринишда ёзилади. Бу мулоҳаза рост бўлиши учун  $A$  ёки  $B$  мулоҳазалардан бири рост бўлиши етарли. Одатда « $\parallel$ » амали «ёки» деб ўқилади.

Юқорида келтирилган фикрлар асосида мантиқий амаллар учун ростлик жадвали аниқланган (1.3-жадвал).

1.3-жадвал. Мантиқий амаллар учун ростлик жадвали

Мулоҳазалар		Мулоҳазалар устида амаллар		
A	B	!A	A && B	A    B
False	false	true	false	false
False	true	true	false	true
True	false	false	false	true
True	true	false	true	true

Мантиқий тур қийматлари устида мантиқий амалларни қўллаш орқали мураккаб мантиқий ифодаларни куриш мумкин. Мисол учун, « $x$  - мусбат ва у қиймати [1..3] сонлар оралиғига тегишли эмас» мулоҳазасини мантиқий ифода кўриниши қуйидашича бўлади:

$$(x > 0) \&\& (y < 1 \parallel y > 3) .$$

**void** тури. *void* туридаги программа объекти ҳеч қандай қийматга эга бўлмайди ва бу турдан қандайдир қурилманинг тил синтаксисига мос келишини таъминлаш учун ишлатилади. Масалан, C++ тили синтаксиси функция қиймат қайтаришини талаб қилади. Агар функция қиймат қайтармайдиган бўлса, яъни унинг қиймат қайтариши зарур бўлмаса, у *void* калит сўзи билан эълон қилинади.

Мисоллар.

```
int a=0,A=1;
float abc=17.5;
double lldiz;
bool Ok=true;
char LETTER='z';
void Mening_Funktsiyam();/* функция қайтарадиган
қиймат инобатга олинмайди */
```

#### НАЗОРАТ ТОПШИРИҚЛАРИ:

1. Ўзгармаслар ва ўзгарувчиларнинг фарқлари
2. Ўзгарувчиларнинг асосий турлари
3. Ростлик жадвали

#### 6-мавзу: Берилганларни компьютер хотирасида тасвирланиши

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

#### ДАРС Р Е Ж А С И (асосий саволлар):

1. C++ тилидаги программа тузилиши ва унинг компиляцияси
2. C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

#### 7-мавзу: Ўзгарувчилар ва ифодалар.

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

#### ДАРС Р Е Ж А С И (асосий саволлар):

1. Ўзгарувчилар турлари
2. Ифода тушунчаси



Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

### Арифметик амаллар. Қиймат бериш оператори

Берилганларни қайта ишлаш учун C++ тилида амалларнинг жуда кенг мажмуаси аниқланган. *Амал* - бу қандайдир ҳаракат бўлиб, у битта (унар) ёки иккита (бинар) операндлар устида бажарилади ва ҳисоб натижаси унинг қайтарувчи қиймати ҳисобланади.

Таянч арифметик амалларга қўшиш ('+'), айириш ('-'), кўпайтириш ('\*'), бўлиш ('/') ва бутун сонли арифметиканинг бўлиш қолдиғини олиш ('%') амалларини келтириш мумкин.

Амаллар қайтарадиган қийматларни ўзлаштириш учун қиймат бериш амали ('=') ва унинг турли модификациялари ишлатилади: қўшиш, қиймат бериш билан ("+="); айириш, қиймат бериш билан ("-="); кўпайтириш, қиймат бериш билан ("\*="); бўлиш, қиймат бериш билан ("/="); бўлиш қолдиғини олиш, қиймат бериш билан ("%=") ва бошқалар. Бу ҳолатларнинг умумий кўриниши:

<ўзгарувчи><амал>=<ифода>;

Қуйидаги программа матнида айрим амалларга мисоллар келтирилган.

```
#include <iostream.h>
int main()
{
    int a=0,b=4,c=90;
    char z='\\t';           // табуляция амали
    a=b; cout<<a<<z;       // a=4
    a=b+c+c+b; cout<<a<<z; // a= 4+90+90+4 = 188
    a=b-2; cout<<a<<z;     // a=2
    a=b*3; cout<<a<<z;     // a=4*3 = 12
    a=c/(b+6); cout<<a<<z; // a=90/(4+6) =9
    cout<<a%2<<z;         // 9%2=1
    a+=b; cout<<a<<z;      // a=a+b = 9+4 =13
    a*=c-50; cout<<a<<z;  //a=a*(c-50)=13*(90-50)=520
    a-=38; cout<<a<<z;    // a=a-38=520-38=482
    a%=8; cout<<a<<z;     // a=a%8=482%8=2
    return 0;
}
```

Программа бажарилиши натижасида экранга қуйидаги сонлар қатори пайдо бўлади:

4      188      2      12      9      1      482      2

## Ифода тушунчаси

C++ тилида *ифода* - амаллар, операндлар ва пунктация белгиларининг кетма-кетлиги бўлиб, компилятор томонидан берилганлар устида маълум бир амалларни бажаришга кўрсатма деб қабул қилинади. Ҳар қандай ‘;’ белги билан тугайдиган ифодага *тил кўрсатмаси* дейилади.

C++ тилидаги тил кўрсатмасига мисол:

```
x=3*(y-2.45) ;  
y=Summa (a , 9 , c) ;
```

**8-мавзу:** Амаллар: инкремент, декремент, sizeof, мантикий, разрядли, таққослаш.

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

1. Инкремент ва декримент
2. Префикс ва постфикс
3. Sizeof буйруғи бажарилиши

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

## Инкремент ва декремент амаллари

C++ тилида бутун турдаги операнд қийматини биттага ошириш ва биттага камайтиришнинг самарали воситалари мавжуд. Булар инкремент (“++”) ва декремент (“--”) унар амалларидир.

Операндга нисбатан бу амалларнинг *префикс* ва *постфикс* кўринишлари бўлади. Префикс кўринишда амал тил кўрсатмаси бўйича иш бажарилишидан олдин операндга қўлланилади. Постфикс ҳолатда эса амал тил кўрсатмаси бўйича иш бажарилгандан кейин операндга қўлланилади.

Префикс ёки постфикс амал тушунчаси қиймат бериш ва бошқа амаллар билан боғлиқ ифодаларда ўринли:

```
x=y++;          // постфикс инкремент  
index =--i;    // префикс декремент  
count++;       // унар амал,  “++count;”   билан  
эквивалент  
abc-- ;        // унар амал,  “--abc;”    билан  
эквивалент
```

Бу ерда у ўзгарувчининг қийматини х ўзгарувчисига ўзлаштирилади ва кейин биттага оширилади, i ўзгарувчининг қиймати биттага

камайтириб, *index* ўзгарувчисига ўзлаштирилади. Ўзгарувчиларнинг аниқ бир қийматлари учун мисол:

```
int s=10,i=5;
s+=++i;      // s=16 ва i=6
s+=i++;      // s=22 ва i=7.
```

### sizeof амали

Ҳар хил турдаги ўзгарувчилар компьютер хотирасида турли сондаги байтлар кетма-кетлигидан иборат жойни эгаллайди. Бунда, ҳаттоки бир турдаги ўзгарувчилар ҳам қайси компьютерда ёки қайси операцион системада амал қилинишига қараб турли ўлчамдаги хотирани банд қилиши мумкин.

C++ тилида ихтиёрий (таянч ва ҳосилавий) турдаги ўзгарувчиларнинг ўлчамини *sizeof* амали ёрдамида аниқланади. Бу амални ўзгармасга, турга ва ўзгарувчига қўллаш мумкин.

Қуйида келтирилган программада компьютернинг платформа-сига мос равишда таянч турларининг ўлчамлари чоп қилинади.

```
int main()
{
    cout<<"int тури ўлчами:"<<sizeof(int)<<' \n' ;
    cout<<"float тури ўлчами:"<<sizeof(float)<<' \n' ;
    cout<<"double тури ўлчами:"<<sizeof(double)
<<' \n' ;
    cout<<"char тури ўлчами:"<<sizeof(char)<<' \n' ;
    return 0;
}
```

### Разрядли мантиқий амаллар

Программа тузиш тажрибаси шуни кўрсатадики, одатда қўйил-ган масалани ечишда бирор ҳолат рўй берган ёки йўқлигини ифода-лаш учун 0 ва 1 қиймат қабул қилувчи байроқлардан фойдаланилади. Бу мақсадда бир ёки ундан ортиқ байтли ўзгарувчилардан фойдаланиш мумкин. Масалан, *bool* туридаги ўзгарувчини шу мақсадда ишлатса бўлади. Бошқа томондан, байроқ сифатида байтнинг разрядларидан фойдаланиш ҳам мумкин. Чунки разрядлар фақат иккита қийматни - 0 ва 1 сонларини қабул қилади. Бир байтда 8 разряд бўлгани учун унда 8 та байроқни кодлаш имконияти мавжуд.

Фараз қилайлик, қўриқлаш тизимига 5 та хона уланган ва тизим тахтасида 5 та чироқча (индикатор) хоналар ҳолатини билдиради: хона қўриқлаш тизими назоратида эканлигини мос индикаторнинг ёниб туриши (разряднинг 1 қиймати) ва хонани тизимга уланмаган-лигини индикатор ўчганлиги (разряднинг 0 қиймати) билдиради. Тизим ҳолатини ифодалаш учун бир байт етарли бўлади ва унинг кичик разрядидан бошлаб бештасини шу мақсадда ишлатиш мумкин:

7	6	5	4	3	2	1	0
			ind5	ind4	ind3	ind2	ind1

Масалан, байтнинг қуйидаги ҳолати 1, 4 ва 5 хоналар кўриқлаш тизимига уланганлигини билдиради:

7	6	5	4	3	2	1	0
x	x	x	1	1	0	0	1

Қуйидаги жадвалда C++ тилида байт разрядлари устида мантикий амаллар мажмуаси келтирилган.

3.1-жадвал. Байт разрядлари устида мантикий амаллар

Амаллар	Мазмуни
&	Мантикий ВА (кўпайтириш)
	Мантикий ЁКИ (қўшиш)
^	Истисно қилувчи ЁКИ
~	Мантикий ИНКОР (инверсия)

Разрядли мантикий амалларнинг бажариш натижаларини жадвал кўринишида кўрсатиш мумкин.

3.2-жадвал. Разрядли мантикий амалларнинг бажариш натижалари

A	B	C=A&B	C=A B	C=A^B	C=~A
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

Юқоридаги келтирилган мисол учун кўриқлаш тизимини ифода-ловчи бир байтли *char* туридаги ўзгарувчини эълон қилиш мумкин:

**char q\_taxtasi=0;**

Бу ерда *q\_taxtasi* ўзгарувчисига 0 қиймат бериш орқали барча хоналар кўриқлаш тизимига уланмаганлиги ифодаланади:

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

Агар 3-хонани тизимга улаш зарур бўлса

**q\_taxtasi=q\_taxtasi|0x04;**

амалини бажариш керак, чунки  $0x04_{16}=00000100_2$  ва мантикий ЁКИ амали натижасида *q\_taxtasi* ўзгарувчиси байти қуйидаги кўринишда бўлади:

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

Худди шундай йўл билан бошқа хоналарни тизимга улаш мумкин, зарур бўлса бирданига иккитасини (зарур бўлса барчасини):

```
q_taxtasi=q_taxtasi|0x1F; // 0x1F16=000111112
```

Мантикий кўпайтириш орқали хоналарни кўриклаш тизимидан чиқариш мумкин:

```
q_taxtasi=q_taxtasi&0xFD; // 0xFD16=111111012
```

Худди шу натижани ‘~’ амалидан фойдаланган ҳолда ҳам олиш мумкин. Иккинчи хона тизимга уланганлиги билдирувчи байт қиймати - 00000010<sub>2</sub>, демак шу ҳолатни инкор қилган ҳолда мантикий кўпайтиришни бажариш керак.

```
q_taxtasi=q_taxtasi&(~0x02); // ~0x0216=111111012
```

Ва ниҳоят, агар 3-хона индикаторини, уни қандай қийматда бўлишидан қатъий назар қарама-қарши ҳолатга ўтказишни *инкор қилувчи ЁКИ* амали ёрдамида бажариш мумкин:

```
q_taxtasi=q_taxtasi^0x04; // 0x0416=000001002
```

Разрядли мантикий амалларни қиймат бериш оператори билан биргаликда бажарилишининг қуйидаги кўринишлари мавжуд:

&= - разрядли *ВА* қиймат бериш билан;

|= - разрядли *ЁКИ* қиймат бериш билан;

^= - разрядли *истисно қилувчи ЁКИ* қиймат бериш билан.

### Чапга ва ўнгга суриш амаллари

Байтдаги битлар қийматини чапга ёки ўнгга суриш учун, мос равишда “<<” ва “>>” амаллари қўлланилади. Амалдан кейинги сон битлар нечта ўрин чапга ёки ўнга суриш кераклигини билдиради.

Масалан:

```
unsigned char A=12; // A=000011002=0x0C16  
A=A<<2; // A=001100002=0x3016=4810  
A=A>>3; // A=000001102=0x0616=610
```

Разрядларни  $n$  та чапга (ўнга) суриш сонни  $2^n$  сонига кўпай-тириш (бўлиш) амали билан эквивалент бўлиб ва нисбатан тез бажа-рилади. Шунинг эътиборга олиш керакки, операнд ишорали сон бўлса, у ҳолда чапга суришда энг чапдаги ишора разряди такрорланади (ишора сақланиб қолади) ва манфий сонлар устида бу амал бажарил-ганда математика нуқтаи-назаридан хато натижалар юзага келади:

```
char B=-120; // B=100010002=0x8816  
B=B<<2; // B=001000002=0x2016=3210  
B=-120; // B=100010002=0x8816  
B=B>>3; // B=111100012=0xF116=-1510
```

Шу сабабли, бу разрядли суриш амаллари ишорасиз (*unsigned*) турдаги қийматлар устида бажарилгани маъқул.

## Таққослаш амаллари

C++ тилида қийматларни солиштириш учун таққослаш амаллари аниқланган (3.3-жадвал). Таққослаш амали бинар амал бўлиб, қуйидаги кўринишга эга:

<операнд<sub>1</sub>> <таққослаш амали> <операнд<sub>2</sub>>

Таққослаш амалларининг натижаси - таққослаш ўринли бўлса, *true* (рост), акс ҳолда *false* (ёлғон) қиймат бўлади. Агар таққослашда арифметик ифода қатнашса, унинг қиймати 0 қийматидан фарқли ҳолатлар учун 1 деб ҳисобланади.

3.3-жадвал. Таққослаш амаллари ва уларнинг қўлланиши

Амаллар	Қўлланиши	Мазмуни (ўқилиши)
<	a<b	“a кичик b”
<=	a<=b	“a кичик ёки тенг b”
>	a>b	“a катта b”
>=	a>=b	“a катта ёки тенг b”
==	a==b	“a тенг b”
!=	a!=b	“a тенг эмас b”

## «Вергул» амали

Тил қурилмаларидаги бир нечта ифодаларни компилятор томонидан яхлит бир ифода деб қабул қилиши учун «вергул» амали қўлланилади. Бу амални қўллаш орқали программа ёзишда маълум бир самарадорликка эришиш мумкин. Одатда «вергул» амали *if* ва *for* операторларида кенг қўлланилади. Масалан, *if* оператори қуйидаги кўринишда бўлиши мумкин:

```
if (i=CallFunc() , i<7) . . .
```

Бу ерда, олдин *CallFunc()* функцияси чақирилади ва унинг натижаси *i* ўзгарувчисига ўзлаштирилади, кейин *i* қиймати 7 билан солиштирилади.

**9-мавзу:** Амалларнинг устунликлари ва бажарилиш йўналишлари.

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

1. Амалларнинг устунликлари
2. Амалларни бажарилиш йўналишлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор, устунлик, йўналиш.

### Амалларнинг устунликлари ва бажарилиш йўналишлари

Анъанавий арифметикадагидек C++ тилида ҳам амаллар маълум бир тартиб ва йўналишда бажарилади. Маълумки, математик ифода-ларда бир хил устунликдаги (приоритетдаги) амаллар учраса (масалан, қўшиш ва айириш), улар чапдан ўнгга бажарилади. Бу тартиб C++ тилидаги ҳам ўринли, бироқ айрим ҳолларда амал ўнгдан чапга бажарилиши мумкин (хусусан, қиймат бериш амалида).

Ифодалар қийматини ҳисоблашда амаллар устунлиги ҳисобга олинади. Биринчи навбатда энг юқори устунликка эга бўлган амал бажарилади.

Қуйидаги жадвалда C++ тилида ишлатиладиган амаллар (операторлар), уларнинг устунлик коэффициентлари ва бажарилиш йўналишлари ( $\leftarrow$  - ўнгдан чапга,  $\Rightarrow$  - чапдан ўнгга) келтирилган.

3.4-жадвал. Амалларнинг устунликлари ва бажарилиш йўналишлари

Оператор	Тавсифи	Устунлик	Йўналиш
::	Кўриниш соҳасига рухсат бериш	16	$\Rightarrow$
[ ]	Массив индекси	16	$\Rightarrow$
( )	Функцияни чақириш	16	$\Rightarrow$
.	Структура ёки синф элементини танлаш	16	$\Rightarrow$
->			
++	Постфикс инкремент	15	$\leftarrow$
--	Постфикс декремент	15	$\leftarrow$
++	Префикс инкремент	14	$\leftarrow$
--	Префикс декремент	14	$\leftarrow$
sizeof	Ўлчамни олиш	14	$\leftarrow$
(<type>)	Турга акслантириш	14	
~	Разрядли мантикий ИНКОР	14	$\leftarrow$
!	Мантикий инкор	14	$\leftarrow$
-	Унар минус	14	$\leftarrow$
+	Унар плюс	14	$\leftarrow$
&	Адресни олиш	14	$\leftarrow$
*	Воситали мурожаат	14	$\leftarrow$
new	Динамик объектни яратиш	14	$\leftarrow$
delete	Динамик объектни йўқ қилиш	14	$\leftarrow$
casting	Турга келтириш	14	
*	Кўпайтириш	13	$\Rightarrow$
/	Бўлиш	13	$\Rightarrow$

%	Бўлиш қолдиғи	13	⇒
+	Қўшиш	12	⇒
-	Айириш	12	⇒
>>	Разряд бўйича ўнгга суриш	11	⇒
<<	Разряд бўйича чапга суриш	11	⇒
<	Кичик	10	⇒
<=	Кичик ёки тенг	10	⇒
>	Катта	10	⇒
>=	Катта ёки тенг	10	⇒
==	Тенг	9	⇒
!=	Тенг эмас	9	⇒
&	Разрядли ВА	8	⇒
^	Разрядли истисно қилувчи ЁКИ	7	⇒
	Разрядли ЁКИ	6	⇒
&&	Мантикий ВА	5	⇒
	Мантикий ЁКИ	4	⇒
?:	Шарт амали	3	⇐
=	Қиймат бериш	2	⇐
*=	Қўпайтириш қиймат бериш билан	2	⇐
/=	Бўлиш қиймат бериш билан	2	⇐
%=	Модулли бўлиш қиймат бериш билан	2	⇐
+=	Қўшиш қиймат бериш билан	2	⇐
- =	Айириш қиймат бериш билан	2	⇐
<<=	Чапга суриш қиймат бериш билан	2	⇐
>>=	Ўнгга суриш қиймат бериш билан	2	⇐
&=	Разрядли ВА қиймат бериш билан	2	⇐
^=	Разрядли истисно қилувчи ЁКИ қиймат бериш билан	2	⇐
=	Разрядли ЁКИ қиймат бериш билан	2	⇐
Throw	Истисно ҳолатни юзага келтириш	2	⇐
,	Вергул	1	⇒

C++ тили программа тузувчисига амалларнинг бажарилиш тартибини ўзгартириш имкониятини беради. Худди математикадаги-дек, амалларни қавслар ёрдамида гуруҳларга жамлаш мумкин. Қавс ишлатишга чеклов йўқ.

Қуйидаги программада қавс ёрдамида амалларни бажариш тартибини ўзгартириш кўрсатилган.

```
#include <iostream.h>
int main()
{
```



```

int x=0,y=0;
int a=3,b=34,c=82;
x=a*b+c;
y=a*(b+c);
cout<<"x= "<<x<<' \n' <<"y= "<<y<<' \n' ;
}

```

Программада амаллар устунлигига кўра  $x$  қийматини ҳисоб-лашда олдин  $a$  ўзгарувчи  $b$  ўзгарувчига кўпайтирилади ва унга  $c$  ўзгарувчи қийматига қўшилади. Навбатдаги кўрсатмани бажаришда эса биринчи навбатда ички қавс ичидаги ифода -  $(b+c)$  қиймати ҳисобланади, кейин бу қиймат  $a$  кўпайтирилиб,  $y$  ўзгарувчисига ўзлаштирилади. Программа бажарилиши натижасида экранга

```

x=184
y=348

```

сатрлари чоп этилади.

## 10-мавзу: Ўқиш-ёзиш оқимлари (cin, cout).

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

ДАРС Р Е Ж А С И (асосий саволлар):

1. Ўқиш оқими
2. Ёзиш оқими

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

### **cout объекти ҳақида қисқача маълумот.**

Кейинги мавзуларда сиз `cout` объектини қандай ишлатиш лозимлигини билиб оласиз. Ҳозир эса у ҳақида қисқача маълумот берамиз. Экрanga маълумотни чиқариш учун `cout` сўзини, ундан сўнг чиқариш операторини (`<<`) киритиш лозим. C++ компилятори (`<<`) белгисини бирта оператор деб қарайди. Қуйидаги листингни таҳлил қиламиз.

#### **2.2. – листинг. `cout` объектини қўлланилиши.**

---

```

1: //2.2.-листинг. cout объектини қўлланилиши
2: # include <iostream.h >

```

```

3: int main()
4: {
5: cout << "Bu son 5 ga teng:" << 5<< "\n";
6: cout <<"endl operatori ekranda yangi";
7: cout << " satrga o`tish amalini bajaradi";
8: cout <<endl;
9: cout << "Bu katta son:\t"<< 70000 <<
10:endl;
11:cout << "Bu 5 va 8 sonlarining yig`indisi:
12:<<\t"<< 8+5 << endl;
13:cout << "Bu kasr son:\t \t"<< (float) 5\8
14:<< endl;
15:cout << "Bu esa juda katta son: \t";
16:cout << (double) 7000*7000 << endl;
17:return 0;
18:};

```

#### НАТИЖА:

```

Bu son 5 ga teng: 5
endl operatori ekranda yangi satrga o`tish
amalini bajaradi
Bu katta son: 70000
Bu 5 va 8 sonlarining yig`indisi: 13
Bu kasr son: 0.625
Bu esa juda katta son: 4.9e+07

```

**Изох**

Айрим компиляторларда cout объектидан кейин математик операцияларни бажариш учун фигурали кавсларни ишлатиш талаб қилинади. У ҳолда 2.2. – листингнинг 11 – сатрида қуйидагича алмаштириш бажариш лозим.

```

11: cout << «Here is the sum of 8
and 5<< (8+5) << endl;

```

**Изох**

endl operatori end line (сатр охири) деган сўздан олинган бўлиб «энд-эл» деб ўкилади.

## Изоҳлар

Сиз дастур ёзаётган вақтингизда нима иш қилмоқчи эканлигингиз доимо аниқ бўлади. Лекин бир ойдан сўнг бу дастурга қайтиш лозим бўлса дастурга тегишли деталлар ва уларнинг вазифалари нимадан иборат эканлигини билмаслигингиз мумкин.

Дастурни бутунлай хотирангиздан ўчириб юбормаслик ва бошқаларга ҳам тушунарли бўлиши учун изоҳлардан фойдаланиш лозим. Изоҳлар компилятор томонидан тушириб қолдириладиган дастурнинг алоҳида сатрида ёки бутун бир блокида қўлланилади. Қуйидаги листингни кўриб чиқамиз.

### 2.3. – листинг. **Salom.cpp** дастури мисолида изоҳларни намоиш қилиш.

---

```
1: # include < iostream.h >
2:
3: main()
4: {
5:     cout << "Salom!\ n";
6: /* бу изоҳ токи изоҳнинг
7: охирини кўрсатувчи белги, яъни юлдузча
8: ва слэш белгиси учрамагунча давом этади */
9: cout << "Bu kommentariy tugadi\ n";
10:// бу изоҳ сатрни охирида тугайди.
11:// Иккита слэшдан сўнг ҳеч қандай текст
12:// бўлмаслиги мумкин.
13: return 0;
14: }
```

---

#### НАТИЖА

Salom

Bu kommentariy tugadi

## 11-мавзу: Операторлар.

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

1. Оператор тушунчаси
2. Операторлар турлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

### Оператор тушунчаси

Программалаш тили операторлари ечилаётган масала алгорит-мини амалга ошириш учун ишлатилади. Операторлар *чизиқли* ва *бошқарув операторларига* бўлинади. Аксарият ҳолатларда оператор-лар «*нуқта-вергул*» (‘;’) белгиси билан тугалланади ва у компилятор томонидан алоҳида оператор деб қабул қилинади (*for* операторининг кавс ичидаги ифодалар бундан мустасно). Бундай оператор *ифода оператори* дейилади. Қиймат бериш амаллари гуруҳи, хусусан, қиймат бериш операторлари ифода операторлари ҳисобланади:

**$I++$ ;  $--j$ ;  $k+=I$ ;**

Программа тузиш амалиётида бўш оператор - ‘;’ ишлатилади. Гарчи бу оператор ҳеч нима бажармаса ҳам, ҳисоблаш ифодаларини тил қурилмаларига мос келишини таъминлайди. Айрим ҳолларда юзага келган «*боши берк*» ҳолатлардан чикиб кетиш имконини беради.

Ўзгарувчиларни эълон қилиш ҳам оператор ҳисобланади ва уларга *эълон оператори* дейилади.

## 12-мавзу: Шарт операторлари

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

1. Шартли ўтиш оператори
2. Тўлиқ шартли ўтиш оператори

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

### Шарт операторлари

Олдинги бобда мисол тариқасида келтирилган программаларда амаллар ёзилиш тартибида кетма-кет ва фақат бир марта бажариладиган

ҳолатлар, яъни чизиқли алгоритмлар келтирилган. Амалда эса камдан-кам масалалар шу тариқа ечилиши мумкин. Аксарият масалалар юзага келадиган турли ҳолатларга боғлиқ равишда мос қарор қабул қилишни (ечимни) талаб этади. C++ тили программанинг алоҳида бўлакларининг бажарилиш тартибини бошқаришга имкон берувчи қурилмаларнинг етарлича катта мажмуа-сига эга. Масалан, программа бажарилишининг бирорта қадамида қандайдир шартни текшириш натижасига кўра бошқарувни програм-манинг у ёки бу бўлагига узатиш мумкин (тармоқланувчи алгоритм). Тармоқланишни амалга ошириш учун шартли оператордан фойдаланилади.

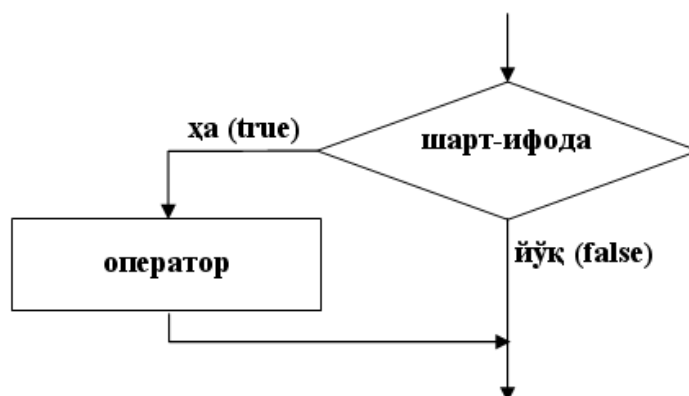
### ***if* оператори**

*if* оператори қандайдир шартни ростликка текшириш натижа-сига кўра программада тармоқланишни амалга оширади:

*if* (<шарт> )<оператор>;

Бу ерда <шарт> ҳар қандай ифода бўлиши мумкин, одатда у таққослаш амали бўлади.

Агар шарт 0 қийматидан фарқли ёки рост (*true*) бўлса, <оператор> бажарилади, акс ҳолда, яъни шарт 0 ёки ёлғон (*false*) бўлса, ҳеч қандай амал бажарилмайди ва бошқарув *if* операторидан кейинги операторга ўтади (агар у мавжуд бўлса). Ушбу ҳолат 4.1-расмда кўрсатилган.



4.1-расм. *if*() шарт операторининг блок схемаси

C++ тилининг қурилмалари операторларни блок кўринишида ташкил қилишга имкон беради. Блок - ‘{’ ва ‘}’ белги оралиғига олинган операторлар кетма-кетлиги бўлиб, у компилятор томонидан яхлит бир оператор деб қабул қилинади. Блок ичида эълон оператор-лари ҳам бўлиши мумкин ва уларда эълон қилинган ўзгарувчилар фақат шу блок ичида кўринади (амал қилади), блокдан ташқарида кўринмайди. Блокдан кейин ‘;’ белгиси қўйилмаслиги мумкин, лекин блок ичидаги ҳар бир ифода ‘;’ белгиси билан якунланиши шарт.

Қуйида келтирилган программада *if* операторидан фойдаланиш кўрсатилган.

```
#include <iostream.h>
```

```

int main()
{
    int b;
    cin>>b;
    if (b>0)
    {
        // b>0 шарт бажарилган ҳолат
        ...
        cout<<"b - musbat son";
        ...
    }
    if (b<0)
        cout<<"b - manfiy son"; // b<0 шарт бажарилган
ҳолат
    return 0;
}

```

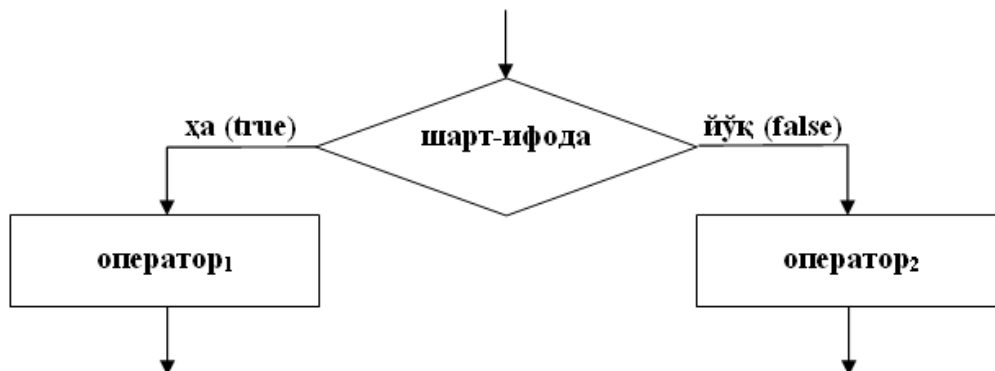
Программа бажарилиши жараёнида бутун турдаги  $b$  ўзгарувчи эълон қилинади ва унинг қиймати клавиатурадан ўқилади. Кейин  $b$  қийматини 0 сонидан катталиги текширилади, агар шарт бажарилса (*true*), у ҳолда ‘{’ ва ‘}’ белгилар ичидаги операторлар бажарилади ва экранга “b - musbat son” хабари чиқади. Агар шарт бажарилмаса, бу операторлар чеклаб ўтилади. Навбатдаги шарт оператори  $b$  ўзгарувчи қиймати манфийликка текширади, агар шарт бажарилса, ягона *cout* кўрсатмаси бажарилади ва экранга “b - manfiy son” хабари чиқади.

### ***if - else* оператори**

Шарт операторининг *if - else* кўриниши қуйидагича:

*if* (<шарт-ифода>) <оператор<sub>1</sub>>; *else* <оператор<sub>2</sub>>;

Бу ерда <шарт-ифода> қиймати 0 қийматидан фарқли ёки *true* бўлса, <оператор<sub>1</sub>>, акс ҳолда <оператор<sub>2</sub>> бажарилади. *if-else* шарт оператори мазмунига кўра алгоритмнинг тармоқланувчи блокини ифода-лайди: <шарт-ифода> - шарт блоки (ромб) ва <оператор<sub>1</sub>> блокнинг «ҳа» шохига, <оператор<sub>2</sub>> эса блокнинг «йўқ» шохига мос келувчи амаллар блоклари деб қараш мумкин (4.2-расм).



4.1-расм. *if()*; *else* шарт операторининг блок схемаси

Мисол тариқасида дискриминантни ҳисоблаш усули ёрдамида  $ax^2+bx+c=0$  кўринишидаги квадрат тенглама илдизларини топиш масаласини кўрайлик:

```
#include <iostream.h>
#include <math.h>
int main()
{
    float a,b,c;
    float D,x1,x2;
    cout<<"\nax^2+bx+c=0  tenglama ildizini topish.
";
    cout<<"\na - koefitsiyentini kiriting: ";
    cin>>a;
    cout<<"\nb - koefitsiyentini kiriting: ";
    cin>>b;
    cout<<"\nc - koefitsiyentini kiriting: ";
    cin>>c;
    D=b*b-4*a*c;
    if (D<0)
    {
        cout<<"\nTenglama haqiqiy ildizga ega emas!";
        return 0;
    }
    if (D==0)
    {
        cout <<"\nTenglama yagona ildizga ega: ";
        x1=-b/(2*a);
        cout<<"\nx= "<<x1;
    }
    else
    {
        cout <<"\nTenglama ikkita ildizga ega: ";
        x1=(-b+sqrt(D))/(2*a);
        x2=(-b-sqrt(D))/(2*a);
        cout<<"\nx1= "<<x1;
        cout<<"\nx2= "<<x2;
    }
    return 0;
}
```

Программа бажарилганда, биринчи навбатда тенглама коэффи-циентлари -  $a$ ,  $b$ ,  $c$  ўзгарувчилар қийматлари киритилади, кейин дискриминант -  $D$  ўзгарувчи қиймати ҳисобланади. Кейин  $D$  қиймати манфийликка текширилади. Агар шарт ўринли бўлса, яхлит оператор сифатида келувчи '{' ва '}' белгилари орасидаги операторлар бажари-лади

ва экранга “Tenglama haqiqiy ildizga ega emas!” хабари чиқади ва программа ўз ишини тугатади (“return 0;” операторини бажариш орқали). Дискриминант нолдан кичик бўлмаса, навбатдаги шарт оператори уни нолга тенглигини текширади. Агар шарт ўринли бўлса, кейинги қаторлардаги операторлар блоки бажарилади - экранга “Tenglama yagona ildizga ega:” хабари, ҳамда  $x1$  ўзгарувчи қиймати чоп этилади ва программа шу ерда ўз ишини тугатади, акс ҳолда, яъни  $D$  қиймати нолдан катта ҳолати учун *else* калит сўзидан кейинги операторлар блоки бажарилади ва экранга “Tenglama ikkita ildizga ega:” хабари, ҳамда  $x1$  ва  $x2$  ўзгарувчилар қийматлари чоп этилади. Шу билан шарт операторидан чиқилади ва асосий функциянинг *return* кўрсатмасини бажариш орқали программа ўз ишини тугатади.

Ўз навбатида <оператор<sub>1</sub>> ва <оператор<sub>2</sub>> ҳам шартли оператор бўлиши мумкин. Ифодадаги ҳар бир *else* калит сўзи, олдиндаги энг яқин *if* калит сўзига тегишли ҳисобланади (худди очилувчи ва ёпилувчи қавслардек). Буни инобатга олмаслик мазмунан хатолик-ларга олиб келиши мумкин.

Масалан:

```
if (x==1)
if (y==1)cout<<"x=1 va y=1";
else cout<<"x<>1";
```

Бу мисолда “ $x<>1$ ” хабари  $x$  қиймати 1 ва  $y$  қиймати 1 бўлмаган ҳолда ҳам чоп этилади. Қуйидаги вариантда ушбу мазмун хатолиги бартараф этилган:

```
if (x==1)
{
if (y==1)cout<<"x=1 va y=1";
}
else cout<<"x<>1";
```

Иккинчи мисол тариқасида учта бутун соннинг максимал қийматини топадиган программа бўлагини келтиришимиз мумкин:

```
...
int x,y,z,max;
cin>>x>>y>>z;
if (x<y)
if (y<z)max=z;
else max=y;
else
if (x<z)max=z;
else max=x;
...
```



Шарт операторида эълон қилиш операторларини ишлатиш ман этилади, лекин ундаги блокларда ўзгарувчиларни эълон қилиш мумкин ва бу ўзгарувчилар фақат блок ичида амал қилади. Қуйидаги мисолда бу ҳолат билан боғлиқ хатолик кўрсатилган:

```
if(j>0){int i;i=2*j;}
else i=-j;//хато,чунки i блокдан ташқарида
кўринмайди
```

**Масала.** Берилган тўрт хонали ишорасиз соннинг бошидаги иккита рақамининг йиғиндиси қолган рақамлар йиғиндисига тенг ёки йўқлиги аниқлансин (рақамлар йиғиндиси деганда уларга мос сон қийматларининг йиғиндиси тушунилади). Соннинг рақамларини ажратиб олиш учун бутун сонлар арифметикаси амалларидан фойдаланилади:

```
#include <iostream.h>
int main()
{
    // n ва унинг  $a_3a_2a_1a_0$  кўринишидаги рақамлар
    эълони
    unsigned int n,a3,a2,a1,a0;
    cout<<"\nn - qiymatini kiriting: ";
    cin>>n;
    if(n<1000||n>9999)
    {
        cout<<"Kiritilgan son 4 xonali emas!";
        return 1;
    }
    a3=n/1000;
    a2=n%1000/100;
    a1=n%100/10;
    a0=n%10;
    if(a3+a2==a1+a0)cout<<"a3+a2 = a1+a0";
    else cout<<"a3+a2<>a1+a0";
    return 0;
}
```

Программа ишорасиз бутун сон киритишни таклиф қилади. Агар киритилган сон 4 хонали бўлмаса ( $n < 1000$  ёки  $n > 9999$ ), бу ҳақда хабар берилади ва программа ўз ишини тугатади. Акс ҳолда  $n$  сонининг рақамлари ажратиб олинади, ҳамда бошидаги иккита рақамнинг йиғиндиси -  $(a_3+a_2)$  қолган иккита рақамлар йиғиндиси -  $(a_1+a_0)$  билан солиштирилади ва уларнинг тенг ёки йўқлиги қараб мос жавоб чоп қилинади.

### 13-мавзу: Танлаш оператори

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

ДАРС Р Е Ж А С И (асосий саволлар):

1. Танлаш оператори
2. Мисоллар

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

### ***switch* оператори**

Шарт операторининг яна бир кўриниши *switch* тармоқланиш оператори бўлиб, унинг синтаксиси қуйидагича:

```
switch (<ифода>)  
{  
    case <ўзгармас ифода1> : <операторлар гуруҳи1>; break;  
    case <ўзгармас ифода2> : <операторлар гуруҳи2>; break;  
    ...  
    case <ўзгармас ифодаn> : <операторлар гуруҳиn>; break;  
    default : <операторлар гуруҳиn+1>;  
}
```

Бу оператор қуйидагича амал қилади: биринчи навбатда <ифода> қиймати ҳисобланади, кейин бу қиймат *case* калит сўзи билан ажратилган <ўзгармас ифода<sub>i</sub>> билан солиштирилади. Агар улар устма-уст тушса, шу қатордаги ‘:’ белгисидан бошлаб, токи *break* калит сўзигача бўлган <операторлар гуруҳи<sub>i</sub>> бажарилади ва бошқарув тармоқланувчи оператордан кейин жойлашган операторга ўтади. Агар <ифода> бирорта ҳам <ўзгармас ифода<sub>i</sub>> билан мос келмаса, қурилманинг *default* қисмидаги <операторлар гуруҳи<sub>n+1</sub>> бажарилади. Шунинг қайд этиш керакки, қурилманинг *default* қисми бўлмаслиги мумкин. Бу ҳолда, агар <ифода> бирорта ҳам *case* ўзгармаси билан мос тушмаса *switch* оператори ҳеч қандай «иш» бажармайди ва бошқарув кейинги операторга ўтади.

Мисол учун, кириш оқимидан “Jarayon davom etilsinmi?” сўровига фойдаланувчи томонидан жавоб олинади. Агар ижобий жавоб олинса, экранга “Jarayon davom etadi!” хабари чоп этилади ва программа ўз ишини тармоқланувчи оператордан кейинги оператор-ларни бажариш билан давом эттиради, акс ҳолда “Jarayon tugadi!” жавоби берилади ва программа ўз ишини тугатади. Бунда, фойдаланувчининг ‘y’ ёки ‘Y’ жавоблари жараёни давом эттиришни билдиради, бошқа белгилар эса жараёни тугатишни англатади.

```
#include <iostream.h>  
int main()
```

```

{
    char Javob=' ';
    cout<<"Jarayon davom etsinmi? ('y','Y') : "
    cin>>Javob;
    switch (Javob)
    {
        case 'Y':
        case 'y':
            cout<<"Jarayon davom etadi!\n";
            break;
        default:
            cout<<"Jarayon tygadi!\n";
            return 0;
    }
    ... // жараён
    return 0;
}

```

Умуман олганда, тармоқланувчи операторда *break* ва *default* калит сўзларини ишлатиш мажбурий эмас. Лекин бу ҳолатда оператор мазмуни бузилиши мумкин. Агар *break* бўлмаса, <ифода> бирорта <ўзгармас ифода> билан устма-уст тушган ҳолда, унга мос келувчи операторлар гуруҳини бажаради ва «*тўхтамасдан*» кейинги қатордаги операторлар гуруҳини ҳам бажаришда давом этади. Масалан, юқоридаги мисолда *break* оператори бўлмаса ва жараёни давом эттиришни тасдиқловчи ('Y') жавоб бўлган тақдирда экранга

```

Jarayon davom etadi!
Jarayon tygadi!

```

хабарлари чиқади ва программа ўз ишини тугатади (*return* операторининг бажарилиши натижасида).

Тармоқланувчи оператор санаб ўтилувчи турдаги ўзгармаслар билан биргаликда ишлатилганда самара беради. Қуйидаги програм-мада ранглар гаммасини тоифалаш масаласи ечилган.

```

#include <iostream.h>
int main()
{
    enum Ranglar{Qizil,Tuq_sariq,Sariq,Yashil,
                  Kuk,Zangori,Binafsha} Rang;

    //...
    switch (Rang)
    {
        case Qizil:
        case Tuq_sariq:
        case Sariq:

```

```

        cout<<"Issiq gamma tanlandi.\n"; break;
    case Yashil:
    case Kuk:
    case Zangori:
    case Binafsha:
        cout<<"Sovuq gamma tanlandi.\n"; break;
    default:cout<<"Kamalak bunday rangga ega
emas.\n";
    }
    return 0;
}

```

Программа бажарилишида бошқарув тармоқланувчи операторга келганда, *Rang* қиймати *Qizil* ёки *Tuq\_sariq* ёки *Sariq* бўлса, “Issiq gamma tanlandi” хабари, агар *Rang* қиймати *Yashil* ёки *Kuk* ёки *Zangori* ёки *Binafsha* бўлса, экранга “Sovuq gamma tanlandi.” хабари, агар *Rang* қиймати санаб ўтилган қийматлардан фарқли бўлса, экранга “Kamalak bunday rangga ega emas.” хабари чоп этилади ва программа ўз ишини тугатади.

*switch* операторида эълон операторлари ҳам учраши мумкин. Лекин *switch* оператори бажарилишида «сакраб ўтиш» ҳолатлари бўлиши ҳисобига блок ичидаги айрим эълонлар бажарилмаслиги ва бунинг оқибатида программа ишида хатолик рўй бериши мумкин:

```

//...
int k=0,n=0;
cin >>n;
switch (n)
{
    int i=10;//хато,бу оператор ҳеч қачон
бажарилмайди
    case 1:
        int j=20;//агар n=2 бўлса,бу эълон бажарилмайди
    case 2:
        k+=i+j; //хато, чунки i,j ўзгарувчилар
номаълум
}
cout<<k;
//...

```

**Масала.** Қуйида санаб ўтилувчи турлар ва шу турдаги ўзгарувчилар эълон қилинган:

```

enum
Birlik{desimetr,kilometr,metr,millimetr,santimetr};
float x; Birlik r;

```

Берилган  $r$  бирликдаги  $x$  ўзгарувчисининг қиймати метрларда чоп қилинсин.

```
#include <iostream.h>
int main()
{
    enum
        Birlik{desimetr,kilometr,metr,millimetr,
santimetr};
    float x,y;
    Birlik r;
    int n;
    cout<<"Uzunlikni kiriting: x=";
    cin>>x;
    cout<<" Uzunlik birliklari\n";
    cout<<" 0- desimetr\n";
    cout<<" 1- kilometr\n";
    cout<<" 2- metr\n";
    cout<<" 3- millimetr\n";
    cout<<" 4- santimetr\n";
    cout<<" Uzunlikni birligini tanlang: r=";
    cin>>n;
    r=n;
    switch(r)
    {
        case desimetr:  y=x/10; break;
        case kilometr:  y=x*1000; break;
        case metr:       y=x; break;
        case millimetr:  y=x/1000; break;
        case santimetr:  y=x/100; break;
        default:
            cout<<"Uzunlik birligi noto'g'ri kiritildi!";
            return 0;
    }
    cout<<y<<" metr";
    return 0;
}
```

**14-мавзу:** for такрорлаш операторлари.

Ажратилган соат:	2 соат
Машғулот тури:	маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

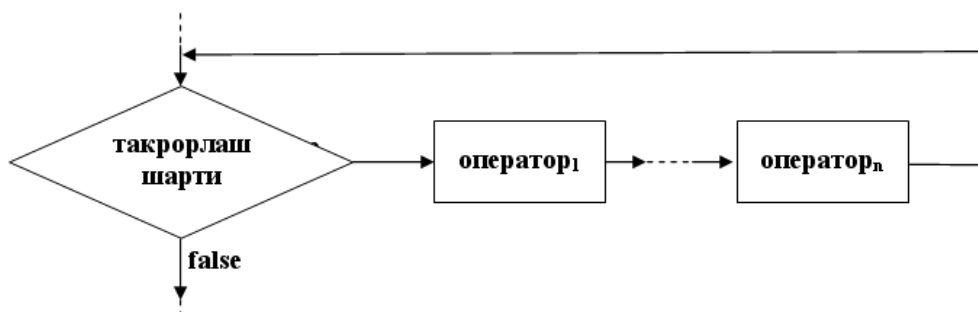
1. Такрорлаш операторлари
2. Параметрли такрорлаш оператори

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

### Такрорлаш операторлари

Программа бажарилишини бошқаришнинг бошқа бир кучли механизмларидан бири - такрорлаш операторлари ҳисобланади.

Такрорлаш оператори «*такрорлаш шарти*» деб номланувчи ифоданинг рост қийматида программанинг маълум бир қисмидаги операторларни (такрорлаш танасини) такрор равишда бажаради (итаратив жараён) (4.2-расм).



4.2-расм. Такрорлаш операторининг блок схемаси

Такрорлаш ўзининг кириш ва чиқиш нуқталарига эга, лекин чиқиш нуқтасининг бўлмаслиги мумкин. Бу ҳолда такрорлашга *чексиз такрорлаш* дейилади. Чексиз такрорлаш учун такрорлашни давом эттириш шарти доимо рост бўлади.

Такрорлаш шартини текшириш такрорлаш танасидаги операторларни бажаришдан олдин текширилиши мумкин (*for*, *while* такрорлашлари) ёки такрорлаш танасидаги операторлари бир марта бажарилгандан кейин текширилиши мумкин (*do-while*).

Такрорлаш операторлари ичма-ич жойлашган бўлиши мумкин.

### *for* такрорлаш оператори

*for* такрорлаш операторининг синтаксиси қўйидаги кўринишга эга:

*for* (<ифода<sub>1</sub>>; <ифода<sub>2</sub>>;<ифода<sub>3</sub>>) <оператор ёки блок>;

Бу оператор ўз ишини <ифода<sub>1</sub>> ифодасини бажаришдан бошлайди. Шундан кейин такрорлаш қадамлари бошланади. Ҳар бир қадамда <ифода<sub>2</sub>> бажарилади, агар натижа 0 қийматидан фарқли ёки *true* бўлса, такрорлаш танаси - <оператор ёки блок> бажарилади ва охири-да <ифода<sub>3</sub>> бажарилади. Агар <ифода<sub>2</sub>> қиймати 0 (*false*) бўлса, такрорлаш жараёни тўхтайди ва бошқарув такрорлаш операторидан кейинги операторга ўтади. Шунини қайд қилиш керакки, <ифода<sub>2</sub>> ифодаси вергул билан ажратилган бир нечта ифодалар бирлашма-сидан иборат бўлиши

мумкин, бу ҳолда охирги ифода қиймати такрорлаш шарти ҳисобланади. Такрорлаш танаси сифатида битта оператор, жумладан бўш оператор бўлиши ёки операторлар блоки келиши мумкин.

Мисол учун 10 дан 20 гача бўлган бутун сонлар йиғиндисини ҳисоблаш масаласини кўрайлик.

```
#include <iostream.h>
int main()
{
    int Summa=0;
    for(int i=10;i<=20;i++)
        Summa+=i;
    cout<<"Yig'indi=" <<Summa;
    return 0;
}
```

Программадаги такрорлаш оператори ўз ишини  $i$  такрорлаш параметрига (такрорлаш санагичига) бошланғич қиймат - 10 сонини беришдан бошлайди ва ҳар бир такрорлаш қадамидан (итарациядан) кейин қавс ичидаги учинчи оператор бажарилиши ҳисобига унинг қиймати биттага ошади. Ҳар бир такрорлаш қадамида такрорлаш танасидаги оператор бажарилади, яъни *Summa* ўзгарувчисига  $i$  қиймати қўшилади. Такрорлаш санагичи  $i$  қиймати 21 бўлганда " $i \leq 20$ " такрорлаш шарти *false* бўлади ва такрорлаш тугайди. Натижада бошқарув такрорлаш операторидан кейинги *cout* операторига ўтади ва экранга йиғинди чоп этилади.

Юқорида келтирилган мисолга қараб такрорлаш операторлари-нинг қавс ичидаги ифодаларига изоҳ бериш мумкин:

<ифода<sub>1</sub>> - такрорлаш санагичи вазифасини бажарувчи ўзгарув-чига бошланғич қиймат беришга хизмат қилади ва у такрорлаш жараёни бошида фақат бир марта ҳисобланади. Ифодада ўзгарувчи эълони учраши мумкин ва бу ўзгарувчи такрорлаш оператори танасида амал қилади ва такрорлаш операторидан ташқарида «*кўринмайди*» (C++ Builder компилятори учун);

<ифода<sub>2</sub>> - такрорлашни бажариш ёки йўқлигини аниқлаб берувчи мантикий ифода, агар шарт рост бўлса, такрорлаш давом этади, акс ҳолда йўқ. Агар бу ифода бўш бўлса, шарт доимо рост деб ҳисобланади;

<ифода<sub>3</sub>> - одатда такрорлаш санагичининг қийматини ошириш (камайтириш) учун хизмат қилади ёки унда такрорлаш шартига таъсир килувчи бошқа амаллар бўлиши мумкин.

Такрорлаш операторида қавс ичидаги ифодалар бўлмаслиги мумкин, лекин синтаксис ';' бўлмаслигига рухсат бермайди. Шу сабабли, энг содда кўринишдаги такрорлаш оператори қуйидагича бўлади:

```
for(;;)cout<<"Cheksiz takrorlash...";
```

Агар такрорлаш жараёнида бир нечта ўзгарувчиларнинг қий-мати синхрон равишда ўзгариши керак бўлса, такрорлаш ифода-ларида зарур операторларни ‘,’ билан ёзиш орқали бунга эришиш мумкин:

```
for(int i=10,j=2;i<=20;i++,j=i+10){...};
```

Такрорлаш операторининг ҳар бир қадамида  $i$  ва  $j$  ўзгарувчи-ларнинг қийматлари мос равишда ўзгариб боради.

*for* операторида такрорлаш танаси бўлмаслиги ҳам мумкин. Масалан, программа бажарилишини маълум бир муддатга «*тўхтаб*» туриш зарур бўлса, бунга такрорлашни ҳеч қандай қўшимча ишларни бажармасдан амал қилиши орқали эришиш мумкин:

```
#include <iostream.h>
int main()
{
    int delay;
    ...
    for(delay=5000;delay>0;delay--); //бўш оператор
    ...
    return 0;
}
```

Юқорида келтирилган 10 дан 20 гача бўлган сонлар йиғинди-сини бўш танали такрорлаш оператори орқали ҳисоблаш мумкин:

```
...
for(int i=10;i<=20;Summa+=i++);
...
```

Такрорлаш оператори танаси сифатида операторлар блоки ишлатишини факториални ҳисоблаш мисолида кўрсатиш мумкин:

```
#include <iostream.h>
int main()
{
    int a;
    unsigned long fact=1;
    cout<<"Butun sonni kiriting:_ ";
    cin>>a;
    if(a>=0 && a<33)
    {
        for(int i=1;i<=a;i++)fact*=i;
        cout<<a<<"!= "<<fact<<' \n' ;
    }
    return 0;
}
```



Программа фойдаланувчи томонидан 0 дан 33 гача ораликдаги сон киритилганда амал қилади, чунки 34! қиймати *unsigned long* учун ажратилган разрядларга сиғмайди.

**Масала.** Такрорлаш операторининг ичма-ич жойлашувига мисол сифатида рақамлари бир-бирига ўзаро тенг бўлмаган уч хонали натурал сонларни ўсиш тартибида чоп қилисин.

```
#include <iostream.h>
int main()
{
    unsigned char a2,a1,a0;          //уч хонали сон
    рақамлари
    for(a2='1';a2<='9';a2++) //соннинг 2-рақами
    for(a1='0';a1<='9';a1++) //соннинг 1-рақами
    for(a0='0';a0<='9';a0++) //соннинг 0-рақами
    // рақамларни ўзаро тенг эмаслигини текшириш
    if(a0!=a1 && a1!=a2 && a0!=a2) //ўзаро тенг
    эмас
        cout<<a2<<a1<<a0<<' \n' ;
    return 0;
}
```

Программада уч хонали соннинг ҳар бир рақами такрорлаш операторларининг параметрлари сифатида ҳосил қилинади. Биринчи, ташқи такрорлаш оператори билан 2-хонадаги рақам (*a2* такрорлаш параметри) ҳосил қилинади. Иккинчи, ички такрорлаш операторида (*a1* такрорлаш параметри) сон кўринишининг 1-хонасидаги рақам ва ниҳоят, унга нисбатан ички бўлган *a0* параметрли такрорлаш операторида 0-хонадаги рақамлар ҳосил қилинади. Ҳар бир ташқи такрорлашнинг бир қадамига ички такрорлаш операторининг тўлиқ бажарилиши тўғри келиши ҳисобига барча уч хонали сонлар кўриниши ҳосил қилинади.

### 15-мавзу: while, do-while такрорлаш операторлари

<b>Ажратилган соат:</b>	2 соат
<b>Машғулот тури:</b>	маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

1. while такрорлаш оператори
2. do-while такрорлаш оператори

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

### ***while* такрорлаш оператори**

*while* такрорлаш оператори, оператор ёки блокни такрорлаш шарти ёлғон (*false* ёки 0) бўлгунча такрор бажаради. У қуйидаги синтаксисга эга:

*while* (<ифода>) <оператор ёки блок>;

Агар <ифода> рост қийматли ўзгармас ифода бўлса, такрорлаш чексиз бўлади. Худди шундай, <ифода> такрорлаш бошланишида рост бўлиб, унинг қийматига такрорлаш танасидаги ҳисоблаш таъсир этмаса, яъни унинг қиймати ўзгармаса, такрорлаш чексиз бўлади.

*while* такрорлаш шартини олдиндан текширувчи такрорлаш оператори ҳисобланади. Агар такрорлаш бошида <ифода> ёлғон бўлса, *while* оператори таркибидаги <оператор ёки блок> қисми бажарилмасдан чеклаб ўтилади.

Айрим ҳолларда <ифода> қиймат бериш оператори кўринишида келиши мумкин. Бунда қиймат бериш амали бажарилади ва натижа 0 билан солиштирилади. Натижа нолдан фарқли бўлса, такрорлаш давом эттирилади.

Агар ифоданинг қиймати нолдан фарқли ўзгармас бўлса, чексиз такрорлаш рўй беради. Масалан:

***while*(1) ; //чексиз такрорлаш**

Худди *for* операторидек, ‘,’ ёрдамида <ифода> таркибида бир нечта амаллар синхрон равишда бажариш мумкин. Масалан, сон ва унинг квадратларини чоп қиладиган программада ушбу ҳолат кўрсатилган:

```
#include <iostream.h>
int main()
{
    int n,n2;
    cout<<"Sonni kiriting(1..10):_";
    cin>>n;
    while (n2=n*n,n)
        cout<<"n="<<n--<<"  n^2="<<n2<<endl;
    return 0;
}
```

Программадаги такрорлаш оператори бажарилишида *n* сони 1 гача камайиб боради. Ҳар бир қадамда *n* ва унинг квадрати чоп қилинади. Шунга эътибор бериш керакки, шарт ифодасида оператор-ларни ёзилиш кетма-кетлигининг аҳамияти бор, чунки энг охирги оператор такрорлаш шарти сифатида қаралади ва *n* қиймати 0 бўлганда такрорлаш тугайди.

Кейинги программада берилган ўнлик соннинг иккилик кўринишини чоп қилиш масаласини ечишда *while* операторини қўллаш кўрсатилган.

```
#include <iostream.h>
int main()
```

```

{
    int sanagich=4;
    short son10,jarayon=1;
    while (jarayon)          // чексиз такрорлаш
    {
        cout <<"\n0'nlik sonni kiriting(0..15) _";
        cin >>son10;
        cout<<' \n'<<son10<<"Sonining                ikkilik
ko'rinishi: ";
        while(sanagich)
        {
            if(son10&8)      //son10 & 00001000
            cout<<'1' ;
            else cout<<'0' ;
            son10=son10<<1;    //разрядлари 1 ўрин чапга
суриш
            sanagich-- ;
        }
        cout<<' \n' ;
        cout<<"Jarayonni to'xtasin(0),davom etsin(1): _
";
        cin>>jarayon;
        sanagich=4;
    }
    return 0;
}

```

Программада ичма-ич жойлашган такрорлаш операторлари ишлатилган. Биринчиси, соннинг иккилик кўринишини чоп қилиш жараёнини давом эттириш шarti бўйича амал қилади. Ички жойлаш-ган иккинчи такрорлаш операторидаги амаллар - ҳар қандай, 0 дан 15 гача бўлган сонлар тўртта разрядли иккилик сон кўринишида бўли-шига асосланган. Унда киритилган соннинг ички, иккилик кўрини-шида учинчи разрядида 0 ёки 1 турганлиги аниқланади (“son10&8”). Шарт натижаси натижа 1 (рост) бўлса, экранга ‘1’, акс ҳолда ‘0’ белгиси чоп этилади. Кейинги кадамда сон разрядлари чапга биттага сурилади ва яна учинчи разряддаги рақам чоп этилади. Такрорлаш *sanagich* қиймати 0 бўлгунча яъни тўрт марта бажарилади ва бошқарув ички такрорлаш операторидан чиқади.

*while* такрорлаш оператори ёрдамида самарали программа коди ёзишга яна бир мисол бу - иккита натурал сонларнинг энг катта умумий бўлувчисини (ЭКУБ) Эвклид алгоритми билан топиш масаласини келтиришимиз мумкин:

```

int main()
{

```

```

int a,b;
cout<<"A va B natural sonlar EKUBini topish.\n";
cout<<"A va B natural sonlarni kiriting: "
cin>>a>>b;
while(a!=b) a>b?a-=b:b-=a;
cout<<"Bu sonlar EKUBi="<<a;
return 0;
}

```

Бутун турдаги  $a$  ва  $b$  қийматлари оқимдан ўқилгандан кейин токи уларнинг қийматлари ўзаро тенг бўлмагунча такрорлаш жараёни рўй беради. Такрорлашнинг ҳар бир қадамида  $a$  ва  $b$  сонларнинг каттасидан кичиги айрилади. Такрорлашдан кейинги кўрсатма воситасида  $a$  ўзгарувчининг қиймати натижа сифатида чоп этилади.

**Масала.** Ўнлик санок системасида берилган  $n$  бутун сонининг иккилик кўринишидаги 1 рақамларининг миқдори аниқлансин. Масалани ечишда разрядли кўпайтириш амалини қўллаган ҳолда ва " $n=(n-1)\&n$ ;" кўрсатмаси  $n$  сонининг иккилик кўринишидаги битта '1' рақамини «ўчиришидан» фойдаланилади. Бу амал токи  $n$  сони 0 бўлгунча давом эттирилади ва такрорлашлар сони жавоб сифатида чоп этилади. Мисол тариқасида  $n=9$  сонидagi 1 рақамлар сонини ҳисоблайлик:

1-қадам:  $n=8_{10}\&9_{10}=1000_2\&1001_2=1000_2=8_{10}$ ;

2-қадам:  $n=7_{10}\&8_{10}=0111_2\&1000_2=0$ .

Такрорлаш сони 2 ва у жавоб бўлади.

Программа матни:

```

#include <iostream.h>
int main()
{
    int n,birlar=0;
    cout<<"O'nlik sonni kiriting:_";
    cin>>n;
    while(n)
    {
        n=(n-1)&n;
        birlar++;
    }
    cout<<"Berilgan          sonning          ikkilik
ko'rinishidagi\n";
    cout<<"birlar soni "<<birlar<<" ta!";
    return 0;
}

```

### ***do-while* такрорлаш оператори**

*do-while* такрорлаш оператори *while* операторидан фарқли равишда олдин оператор ёки блокни бажаради, кейин такрорлаш шартини

текширади. Бу қурилма такрорлаш танасини камида бир марта бажарилишини таъминлайди. *do-while* такрорлаш оператори қуйидаги синтаксисга эга:

*do* <оператор ёки блок>; *while* (<ифода>);

Бундай такрорлаш операторининг кенг қўлланиладиган ҳолатлар - бу такрорлашни бошламасдан туриб, такрорлаш шартини текширишнинг иложи бўлмаган ҳолатлар ҳисобланади. Масалан, бирорта жараёни давом эттириш ёки тугатиш ҳақидаги сўровга жавоб олиш ва уни текшириш зарур бўлсин. Кўриниб турибдики, жараёни бошламасдан олдин бу сўровни беришнинг маъноси йўқ. Ҳеч бўлмаганда такрорлаш жараёнининг битта қадами амалга оширилган бўлиши керак:

```
#include <iostream.h>
int main()
{
    char javob;
    do
    {
        ...// амаллар кетма-кетлиги
        cout<<"Jarayonni to'xtatish (N):_ ";
        cin>>javob;
    }
    while(javob!=N)
    return 0;
}
```

Программа токи "Jarayonni to'xtatish (N):\_ " сўровига 'N' жавоби киритилмагунча давом этади.

Бу оператор ҳам чексиз такрорланиши мумкин:

**do; while(1);**

**Масала.** Ҳар қандай 7 сонидан катта бутун сондаги пул миқдорини 3 ва 5 сўмликларда бериш мумкинлиги исботлансин. Қўйилган масала  $p=3n+5m$  тенгламаси қаноатлантирувчи  $m, n$  сонлар жуфтликларини топиш масаласидир ( $p$ -пул миқдори). Бу шартнинг бажарилишини  $m$  ва  $n$  ўзгарувчиларининг мумкин бўлган қийматларининг барча комбинацияларида текшириш зарур бўлади.

```
#include <iostream.h>
int main()
{
    unsigned int Pul; //Pul- киритиладиган пул
миқдори
    unsigned n3,m5;    //n-3 сўмликлар,m-5 сўмликлар
сони
```

```

    bool xato=false; //Pul қийматини киритишдаги
хатолик
    do
    {
        if(xato)cout<<"Pul qiymati 7 dan kichik!";
        cout<<"\nPul qiymatini kiriting (>7): ";
        cin>>Pul;
        xato=true; //хато ҳақида хабар бериш байроғи
    }
    while(Pul<=7); //токи 7 катта сон киритилгунча
n3=0; //бирорта ҳам 3 сўмлик йўқ
do
{
    m5=0; //бирорта ҳам 5 сўмлик йўқ
    do
    {
        if (3*n3+5*m5==Pul)
            cout<<n3<<" ta 3 so'mlik+"<<m5<<" ta 5
so'mlik\n";
        m5++; // 5 сўмликлар биттага оширилади
    }
    while(3*n3+5*m5<=Pul);
    n3++; //3 сўмликлар биттага оширилади
}
while(3*n3<=Pul);
return 0;
}

```

Программа пул қийматини киритишни сўрайди (*Pul* ўзгарувчи-сига). Агар пул қиймати 7 сонидан кичик бўлса, бу ҳақда хабар берилади ва такрор равишда қиймат киритиш талаб қилинади. Пул қиймати 7 дан катта бўлганда, 3 ва 5 сўмликларнинг мумкин бўлган тўла комбинациясини амалга ошириш учун ичма-ич такрорлашлар амалга оширилади. Ташқи такрорлаш *n3* (3 сўмликлар миқдори) бўйича, ички такрорлаш эса *m5* (5 сўмликлар миқдори) бўйича, токи бу миқдордаги пуллар қиймати *Pul* қийматидан ошиб кетмагунча давом этади. Ички такрорлашда *m5* ўзгарувчисининг ҳар бир қийматида " $3*n3+5*m5==Pul$ " шарти текширилади, агар у ўринли бўлса, ечим варианты сифатида *n3* ва *m5* ўзгарувчилар қийматлари чоп этилади.

Пул қиймати 30 сўм киритилганда (*Pul=30*), экранга

```

0 ta 3 so'mlik + 6 ta 5 so'mlik
5 ta 3 so'mlik + 3 ta 5 so'mlik
10 ta 3 so'mlik + 0 ta 5 so'mlik

```

ечим вариантлари чоп этилади.

## 16-мавзу: Бошқарувни узатиш операторлари

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

1. break оператори
2. continue оператори
3. goto оператори

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

### ***break* оператори**

Такрорлаш операторларининг бажарилишида шундай ҳолатлар юзага келиши мумкинки, унда қайсидир қадамда, такрорлашни якунига етказмасдан такрорлашдан чиқиш зарурати бўлиши мумкин. Бошқача айтганда, такрорлашни «узиш» керак бўлиши мумкин. Бунда *break* операторидан фойдаланилади. *break* операторини такрорлаш оператори танасининг ихтиёрий (зарур) жойларига қўйиш орқали шу жойлардан такрорлашдан чиқишни амалга ошириш мумкин. Эътибор берадиган бўлсак, *switch-case* операторининг туб моҳиятига ҳам *break* операторини қўллаш орқали эришилган.

Ичма - ич жойлашган такрорлаш ва *switch* операторларида *break* оператори фақат ўзи жойлашган блокдан чиқиш имкониятини беради.

Қуйидаги программада иккита ичма-ич жойлашган такрорлаш операторидан фойдаланган ҳолда фойдаланувчи томонидан киритилган қандайдир сонни 3 ва 7 сонларига нисбатан қандай ораликқа тушиши аниқланади. Ташқи такрорлашда “Son kiriting (0- to'xtash):\_” сўрови берилади ва жавоб *javob\_son* ўзгарувчисига ўқилади. Агар сон нолдан фарқли бўлса, ички такрорлаш операторида бу соннинг қандайдир ораликқа тушиши аниқланиб, шу ҳақида хабар берилади ва ички такрорлаш операторидан чиқилади. Ташқи такрорлашдаги сўровга жавоб тариқасида 0 киритилса, программа ўз ишини тугатади.

```
#include <iostream.h>
int main()
{
    int javob_son=0;
    do
    {
        while(javob_son)
        {
            if(javob_son<3){cout<<"3 kichik !"; break;}
        }
    }
}
```

```

    if(3<=javob_son && javob_son<=7)
        {cout<<"3 va 7 oraligida !"; break;}
    if(javob_son>7){cout<<"7 dan katta !"; break;}
}
cout<<"\nSon kiriting (0-to'xtash):_";
cin>>javob_son;
}
while(javob_son!=0)
return 0;
}

```

Амалиётда *break* операторидан чексиз такрорлашдан чиқишда фойдаланилади.

```

for (;;)
{
    // 1- шарт
    if(...){... break; }
    // 2- шарт
    if(...){... break; }
    ...
}

```

Бу мисолда чексиз *for* такрорланишидан 1 ёки 2 - шарт бажарилганда чиқилади.

**Масала.** Ишорасиз бутун сонлар кетма-кетлигининг камаймай-диган ҳолда тартибланган ёки йўқлиги аниқлансин. Программа, кетма-кетликнинг навбатдаги ҳади сифатида 0 қиймати киритилганда (0 кетма-кетлик ҳади ҳисобланмайди) ёки камаймаслик тартибини бузадиган ҳад киритилганда тўхтасин.

```

#include <iostream.h>
int main()
{
    unsigned int a1,a2;
    cout<<"Sonlar ketma-ketligini kiriting";
    cout<<"(0-tugash alomati):\n";
    cin>>a1; // кетма-кетликнинг биринчи ҳади
    while(a1)
    {
        cin>>a2; // навбатдаги ҳад
        if(a2==0||a1>a2)break; //кетма-кетлик тугади
        //тартибланганлик бузилди
        a1=a2;
    }
    if(a1) // камида битта ҳад мавжуд

```

ёки



```

{
    cout<<"Ketma-ketlik tartiblangan";
    if(!a2)cout<<"!"; // кетма-кетлик 0 тугади
    else cout<<" emas!"; // тартиб бузилган
}
else cout<<"Ketma-ketlik bo'sh!";
return 0;
}

```

Бошда кетма-кетликнинг биринчи ҳади алоҳида ўқиб олинади (*a1* ўзгарувчисига). Кейин *a1* нолга тенг бўлмагунча такрорлаш оператори амал қилади. Такрорлаш танасида кетма-кетликнинг навбатдаги ҳади *a2* ўзгарувчисига ўқилади ва *a1* қиймати билан солиштирилади. Агар *a2* нолга тенг ёки у *a1* қийматидан кичик, *break* оператори ёрдамида такрорлаш жараёни узилади ва бошқарув такрорлашдан кейинги шарт операторига ўтади. Бу ердаги шарт операторлари мазмуни қуйидагича: агар *a1* нолдан фарқли бўлса, кетма-кетликнинг камида битта ҳади киритилган бўлади (кетма-кетлик мавжуд) ва охириги киритилган ҳад текширилади. Ўз навбатида агар *a2* нолдан фарқли бўлса, бу ҳолат ҳадлар ўртасида камаймаслик шарти бузилганлигини ва шу сабабли ҳадларни киритиш жараёни узилганини билдиради, бу ҳақда хабар чоп этилади. Акс ҳолда кетма-кетликни камаймайдиган ҳолда тартибланган бўлади. Агар *a1* қиймати нолга тенг бўлса "Ketma-ketlik bo'sh!" хабари берилади ва программадан чиқилади.

### ***continue* оператори**

*continue* оператори худди *break* операторидек такрорлаш оператори танасини бажаришни тўхтатади, лекин такрорлашдан чиқиб кетмасдан кейинги қадамига «сақраб» ўтишини тайинлайди.

*continue* операторини қўлланишига мисол тариқасида 2 ва 50 сонлар оралиғидаги тўб сонларни топадиган программа матнини келтирамыз.

```

#include <iostream.h>
int main()
{
    bool bulinadi=false;
    for(int i=2;i<50;i++)
    {
        for(int j=2;j<i/2;j++)
        {
            if(i%j)continue;
            bulinadi=true;
            break;
        }
        // break бажарилганда бошқарув ўтадиган жой
        if(!bulinadi)cout<<i<<' ' ;
    }
}

```

```

    bulinadi=false;
}
    return 0;
}

```

Келтирилган программада қўйилган масала ичма-ич жойлашган иккита такрорлаш операторлари ёрдамида ечилган. Биринчи такрор-лаш оператори 2 дан 50 гача сонларни ҳосил қилишга хизмат қилади. Ички такрорлаш эса ҳар бир ҳосил қилинаётган сонни 2 сонидан токи шу соннинг ярмигача бўлган сонларга бўлиб, қолдиғини текширади, агар қолдиқ 0 сонидан фарқли бўлса, навбатдаги сонга бўлиш давом этади, акс ҳолда *bulinadi* ўзгарувчисига *true* қиймат бериб, ички такрорлаш узилади (сон ўзининг ярмигача бўлган қандайдир сонга бўлинар экан, демак у туб эмас ва кейинги сонларга бўлиб текширишга ҳожат йўқ). Ички *j* бўйича такрорлашдан чиққандан кейин *bulinadi* қиймати *false* бўлса (*!bulinadi*), *i* сони туб бўлади ва у чоп қилинади.

### **goto оператори ва нишонлар**

*Нишон* - бу давомида иккита нуқта (':') қўйилган идентифи-катор. Нишон билан қандайдир оператор белгиланади ва кейинчалик, программанинг бошқа бир қисмидан унга шартсиз ўтиш амалга оширилади. Нишон билан ҳар қандай оператор белгиланиши мумкин, шу жумладан эълон оператори ва бўш оператори ҳам. Нишон фақат функциялар ичида амал қилади.

Нишонга шартсиз ўтиш *goto* оператори ёрдамида бажарилади. *goto* оператори орқали фақат унинг ўзи жойлашган функция ичидаги операторларга ўтиш мумкин. *goto* операторининг синтаксиси қуйида-гича:

```
goto <нишон>;
```

Айрим ҳолларда, *goto* операторининг «*сакраб ўтиши*» ҳисобига хатоликлар юзага келиши мумкин. Масалан,

```

int i=0;
i++;
if(i)goto m;
int j;
m:j+=i;

```

операторларининг бажарилиши хатоликка олиб келади, чунки *j* эълон қилинмай қолади.

Шартсиз ўтиш оператори программани тузишдаги кучли ва шу билан биргаликда хавфли воситалардан бири ҳисобланади. Кучлилиги шундаки, унинг ёрдамида алгоритмнинг «*боши берк*» жойларидан чиқиб кетиш мумкин. Иккинчи томондан, блокларнинг ичига ўтиш, масалан, такрорлаш операторларини ичига «*сакраб*» кириш кутилма-ган ҳолатларни юзага келтириши мумкин. Шу сабабли, имкон қадар *goto* операторидан фойдаланмаслик керак, ишлатилган тақдирда ҳам қуйидаги қоидага амал

қилиш зарур: «блок ичига, *if...else* ва *switch* операторлари ичига, ҳамда такрорлаш операторлари танасига таш-қаридан кириш мумкин эмас».

Гарчи, нишон ёрдамида программанинг ихтиёрий жойига ўтиш мумкин бўлса ҳам, бошланғич қиймат бериш эълонларидан сакраб ўтиш ман этилади, лекин блоклардан сакраб ўтиш мумкин.

Масалан:

```
...
goto B;          \\ хато
float x=0.0;
goto B;          \\ тўғри
{int n=10;x=n*x+x;}
B:cout<<"x="<<x;
...
```

Хусусан, нишон ёрдамида ички блокдан ташқи блокка ва ташқи блокдан ички блокка ўтишга C++ тили рухсат беради:

```
{...
goto ABC:
...
{
    int i=15;
    ...
    ABC:
    ...
    goto XYZ;
    int y=10;
    ...
    XYZ:
    ...
    goto KLM;
    ...
}
...
int k=0;
...
KLM:
...
}
```

Лекин, юқорида келтирилган мисолдаги биринчи ўтиш (“goto ABC”) мазмунан хато ҳисобланади.

Қуйидаги программада иккита натурал сонлар энг катта умумий бўлувчини (ЭКУБ) топиш масаласидаги такрорлаш жараёнини нишон ва *goto* оператори воситасида амалга ошириш кўрсатилган:

```
int main()
```

```

{
    int a,b;
    cout<<"A va B natural sonlar EKUBini topish.\n";
    cout<<"A va B natural sonlarni kiriting: "
    cin>>a>>b;
    nishon:
    if(a==b)
    {
        cout<<"Bu sonlar EKUBi: "<<a;
        return 0;
    }
    a>b?a-=b:b-=a;
    goto nishon;
}

```

Программадаги нишон билан белгиланган операторда  $a$  ва  $b$  сонларни тенглиги текширилади. Агар улар тенг бўлса, ихтиёрий биттаси, масалан  $a$  сони ЭКУБ бўлади ва функциядан чиқилади. Акс ҳолда, бу сонларнинг каттасидан кичиги айрилади ва *goto* орқали уларнинг тенглиги текширилади. Такрорлаш жараёни  $a$  ва  $b$  сонлар ўзаро тенг бўлгунча давом этади.

Шуни қайд этиш керакки, бу масалани такрорлаш операторлари ёрдамида бажариш анча самарали ҳисобланади.

## 17-мавзу: Структуралар дастурлаш

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

ДАРС Р Е Ж А С И (асосий саволлар):

- 1.Структуралар
- 2.Структура ҳосил қилиш

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

### Структуралар

Маълумки, бирор предмет соҳасидаги масалани ечишда ундаги объектлар бир нечта, ҳар хил турдаги параметрлар билан аниқланиши мумкин. Масалан, текисликдаги нуқта ҳақиқий турдаги  $x$  - абцисса ва  $y$  - ордината жуфтлиги -  $(x,y)$  кўринишида берилади. Талаба ҳақидаги маълумотлар: сатр туридаги талаба фамилия, исми ва шарифи,

мутахассислик йўналиш, талаба яшаш адреси, бутун турдаги туғилган йили, ўқув босқичи, ҳақиқий турдаги рейтинг бали, мантиқий турдаги талаба жинси ҳақидаги маълумот ва бошқалардан шаклланади.

Программада ҳолат ёки тушунчани тавсифловчи ҳар бир берилганлар учун алоҳида ўзгарувчи аниқлаб масалани ечиш мумкин. Лекин бу ҳолда объект ҳақидаги маълумотлар «тарқоқ» бўлади, уларни қайта ишлаш мураккаблашади, объект ҳақидаги берилган-ларни яхлит ҳолда кўриш қийинлашади.

C++ тилида бир ёки ҳар хил турдаги берилганларни жамланмаси *структура* деб номланади. Структура фойдаланувчи томонидан аниқланган берилганларнинг янги тури ҳисобланади. Структура куйидагича аниқланади:

```
struct <структура номи>
{
    <тур1> <ном1>;
    <тур2> <ном2>;
    ...
    <турn> <номn>;
};
```

Бу ерда <структура номи> - структура кўринишида яратилаётган янги турнинг номи, “<тур<sub>i</sub>> <ном<sub>i</sub>>;” - структуранинг *i*-майдонининг (ном<sub>i</sub>) эълони.

Бошқача айтганда, структура эълон қилинган ўзгарувчилардан (майдонлардан) ташкил топади. Унга ҳар хил турдаги берилганларни ўз ичига олувчи *қобик* деб қараш мумкин. Қобикдаги берилганларни яхлит ҳолда кўчириш, ташқи қурилмалар (бинар файлларга) ёзиш, ўқиш мумкин бўлади.

Талаба ҳақидаги берилганларни ўз ичига олувчи структура турининг эълон қилинишини кўрайлик.

```
struct Talaba
{
    char FISH[30];
    unsigned int Tug_yil;
    unsigned int Kurs;
    char Yunalish[50];
    float Reyting;
    unsigned char Jinsi[5];
    char Manzil[50];
    bool status;
};
```

Программада структуралардан фойдаланиш, шу турдаги ўзгарувчилар эълон қилиш ва уларни қайта ишлаш орқали амалга оширилади:

```
Talaba talaba;
```

Структура турини эълонида турнинг номи бўлмаслиги мумкин, лекин бу ҳолда структура аниқланишидан кейин албатта ўзгарувчилар номлари ёзилиши керак:

```
struct
{
    unsigned int x,y;
    unsigned char Rang;
} Nuqta1, Nuqta2;
```

Келтирилган мисолда структура туридаги *Nuqta1*, *Nuqta2* ўзгарувчилари эълон қилинган.

Структура туридаги ўзгарувчилар билан ишлаш, унинг майдон-лари билан ишлашни англатади. Структура майдонига мурожаат қилиш ‘.’ (нуқта) орқали амалга оширилади. Бунда структура тури-даги ўзгарувчи номи, ундан кейин нуқта қўйилади ва майдон ўзгарув-чисининг номи ёзилади. Масалан, талаба ҳақидаги структура майдон-ларига мурожаат қуйидагича бўлади:

```
talaba.Kurs=2;
talaba.Tug_yil=1988;
strcpy(talaba.FISh,"Abdullaev A.A.");
strcpy(talaba.Yunalish,
"Informatika va Axborot texnologiyalari");
strcpy(talaba.Jinsi,"Erk");
strcpy(talaba.Manzil,
"Toshkent,Yunusobod 6-3-8, tel: 224-45-78");
talaba.Reyting=123.52;
```

Келтирилган мисолда *talaba* структурасининг сон туридаги майдонларига оддий кўринишда қийматлар берилган, сатр туридаги майдонлар учун *strcpy* функцияси орқали қиймат бериш амалга оширилган.

Структура туридаги объектнинг хотирадан қанча жой эгаллаганлигини *sizeof* функцияси (оператори) орқали аниқлаш мумкин:

```
int i=sizeof(Talaba);
```

Айрим ҳолларда структура майдонлари ўлчамини разрядларда аниқлаш орқали эгалланадиган хотирани камайтириш мумкин. Бунинг учун структура майдони қуйидагича эълон қилинади:

<майдон номи> : <ўзгармас ифода>

Бу ерда <майдон номи>- майдон тури ва номи, <ўзгармас ифода>- майдоннинг разряддаги узунлиги. Майдон тури бутун турлар бўлиши керак (*int*, *long*, *unsigned*, *char*).

Агар фойдаланувчи структуранинг майдони фақат 0 ва 1 қийма-тини қабул қилишини билса, бу майдон учун бир бит жой ажратиши мумкин (бир байт ёки икки байт ўрнига). Хотирани тежаш эвазига майдон устида амал бажаришда разрядли арифметикани қўллаш зарур бўлади.

Мисол учун сана-вақт билан боғлиқ структурани яратишнинг иккита вариантыни кўрайлик. Структура йил, ой, кун, соат, минут ва секунд майдонларидан иборат бўлсин ва уни қуйидагича аниқлаш мумкин:

```
struct Sana_vaqt
{
    unsigned short Yil;
    unsigned short Oy;
    unsigned short Kun;
    unsigned short Soat;
    unsigned short Minut;
    unsigned short Sekund;
};
```

Бундай аниқлашда *Sana\_vaqt* структураси хотирада *6 майдон\*2 байт=12 байт* жой эгаллайди. Агар эътибор берилса структурада ортиқча жой эгалланган ҳолатлар мавжуд. Масалан, йил учун қиймати 0 сонидан 99 сонигача қиймат билан аниқланиши етарли (масалан, 2011 йилни 11 қиймати билан ифодалаш мумкин). Шунинг учун унга 2 байт эмас, балки 7 разряд ажратиш етарли. Худди шундай ой учун [1..12] қийматларини ифодалашга 4 разряд жой етарли ва ҳакоза.

Юқорида келтирилган чекловлардан кейин сана-вақт структура-сини тежамли вариантыни аниқлаш мумкин:

```
struct Sana_vaqt2
{
    unsigned Yil:7;
    unsigned Oy:4;
    unsigned Kun:5;
    unsigned Soat:6;
    unsigned Minut:6;
    unsigned Sekund:6;
};
```

Бу структура хотирадан 5 байт жой эгаллайди, лекин шуни ҳисобига қийматларни қайта ишлашда разрядли амалларни қўллашга тўғри келади.

### 18-мавзу: Бир ўлчовли статик массивлар.

<b>Ажратилган соат:</b>	2 соат
<b>Машғулоти тури:</b>	маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

- 1.Массив тушунчаси
- 2.Бир ўлчовли статик массивлар

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

### Берилганлар массиви тушунчаси

Хотирада кетма-кет (регуляр) жойлашган бир хил турдаги қийматларга *массив* дейилади.

Одатда массивларга зарурат, катта ҳажмдаги, лекин чекланган миқдордаги ва тартибланган қийматларни қайта ишлаш билан боғлиқ масалаларни ечишда юзага келади. Фараз қилайлик, талабалар гуруҳининг рейтинг баллари билан ишлаш масаласи қўйилган. Унда гуруҳнинг ўртача рейтингини аниқлаш, рейтингларни камайиши бўйича тартиблаш, конкрет талабанинг рейтинг ҳақида маълумот бериш ва бошқа масала остиларини ечиш зарур бўлсин. Қайд этилган масалаларни ечиш учун берилганларнинг (рейтингларнинг) тартиб-ланган кетма-кетлиги зарур бўлади. Бу ерда тартибланганлик маъноси шундаки, кетма-кетликнинг ҳар бир қиймати ўз ўрнига эга бўлади (биринчи талабанинг рейтинг массивда биринчи ўринда, иккинчи талабаники - иккинчи ўринда ва ҳакоза). Берилганлар кетма-кет-лигини икки хил усулда ҳосил қилиш мумкин. Биринчи йўл - ҳар бир рейтинг учун алоҳида ўзгарувчи аниқлаш:  $Reyting_1, \dots, Reyting_N$ . Лекин, гуруҳдаги талабалар сони етарлича катта бўлганда, бу ўзгарувчилар қатнашган программани тузиш катта қийинчиликларни юзага келтиради. Иккинчи йўл - берилганлар кетма-кетлигини ягона ном билан аниқлаб, унинг қийматларига мурожаатни, шу қийматларнинг кетма-кетликда жойлашган ўрнининг номери (индекси) орқали амалга оширишдир. Рейтинглар кетма-кетлигини *Reyting* деб номлаб, ундаги қийматларига  $Reyting_1, \dots, Reyting_N$  кўринишида мурожаат қилиш мумкин. Одатда берилганларнинг бундай кўринишига массивлар дейилади. Массивларни математикадаги сонлар векторига ўхшатиш мумкин, чунки вектор ҳам ўзининг индивидуал номига эга ва у фиксирланган миқдордаги бир турдаги қийматлардан - сонлардан иборатдир.

Демак, *массив* - бу фиксирланган миқдордаги айрим қийматлар-нинг (массив элементларининг) тартибланган мажмуасидир. Барча элементлар бир хил турда бўлиши керак ва бу тур *элемент тури* ёки массив учун *таянч тур* деб номланади. Юқоридаги келтирилган мисолда *Reyting* - ҳақиқий турдаги *вектор* деб номланади.

Программада ишлатиладиган ҳар бир конкрет массив ўзининг индивидуал номига эга бўлиши керак. Бу номни *тўлиқ ўзгарувчи* дейилади, чунки унинг қиймати массивнинг ўзи бўлади. Массивнинг ҳар бир элементи массив номи, ҳамда квадрат қавсга олинган ва *элемент селектори* деб номланувчи индексни кўрсатиш орқали ошкор равишда белгиланади. Мурожаат синтаксиси:

<массив номи >[<индекс>]

Бу кўринишга *хусусий ўзгарувчи* дейилади, чунки унинг қиймати массивнинг алоҳида элементидир. Юқоридаги мисолда *Reyting* массиви-нинг



алоҳида компоненталарига  $Reyting[1], \dots, Reyting[N]$  хусусий ўзгарувчилар орқали мурожаат қилиш мумкин. Бошқача бу ўзгарувчилар *индексли ўзгарувчилар* дейилади.

Массив индекси сифатида бутун сон қўлланилади. Умуман олганда индекс сифатида бутун сон қийматини қабул қиладиган ихтиёрий ифода ишлатилиши мумкин ва унинг қиймати массив элементининг тартиб номерини аниқлайди. Ифода сифатида ўзгарувчи ҳам олиниши мумкинки, ўзгарувчининг қиймати ўзгариши билан мурожаат қилинаётган массив элементини аниқловчи индекс ҳам ўзгаради. Шундай қилиб, программадаги биттагина индексли ўзгарувчи воситасида массивнинг барча элементларини белгилаш (аниқлаш) мумкин бўлади. Масалан,  $Reyting[I]$  ўзгарувчиси орқали  $I$  ўзгарувчи-нинг қийматига боғлиқ равишда  $Reyting$  массивининг ихтиёрий элементиға мурожаат қилиш мавжуд.

Ҳақиқий турдаги (*float*, *double*) қийматлар тўплами чексиз бўлганлиги сабабли улар индекс сифатида ишлатилмайди.

C++ тилида индекс доимо 0 дан бошланади ва унинг энг катта қиймати массив эълонидаги узунликдан биттага кам бўлади.

Массив эълони қуйидагича бўлади:

$\langle \text{тур} \rangle \langle \text{ном} \rangle [\langle \text{узунлик} \rangle] = \{ \text{бошланғич қийматлар} \}.$

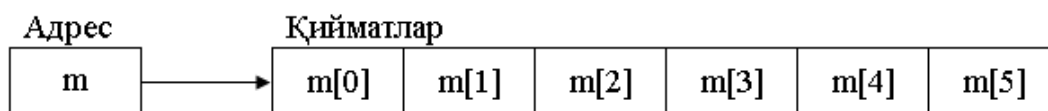
Бу ерда  $\langle \text{узунлик} \rangle$  - ўзгармас ифода. Мисоллар:

```
int m[6]={1,4,-5,2,10,3};
float a[4];
```

Массив статик ва динамик бўлиши мумкин. Статик массивнинг узунлиги олдиндан маълум бўлиб, у хотирада маълум адресдан бошлаб кетма-кет жойлашади. Динамик массивни узунлиги прог-рамма бажарилиш жараёнида аниқланиб, у динамик хотирадаги айни пайтда бўш бўлган адресларға жойлашади. Масалан,

```
int m[6];
```

кўринишида эълон қилинган бир ўлчамли массив элементлари хотирада қуйидагича жойлашади:



7.1-расм. Бир ўлчамли массивнинг хотирадаги жойлашуви

Массивнинг  $i$ - элементиға  $m[i]$  ёки  $*(m+i)$  - воситали мурожаат қилиш мумкин. Массив узунлигини  $sizeof(m)$  амали орқали аниқлади.

Массив эълонида унинг элементларига бошланғич қийматлар бериш мумкин ва унинг бир нечта вариантлари мавжуд.

1) ўлчами кўрсатилган массив элементларини тўлиқ инициализациялаш:

```
int t[5]={-10,5,15,4,3};
```

Бунда 5 та элементдан иборат бўлган  $t$  номли бутун турдаги бир ўлчамли массив эълон қилинган ва унинг барча элементларига бошланғич қийматлар берилган. Бу эълон қуйидаги эълон билан эквивалент:

```
int t[5];  
t[0]=-10; t[1]=5; t[2]=15; t[3]=4; t[4]=3;
```

2) ўлчами кўрсатилган массив элементларини тўлиқмас инициализациялаш:

```
int t[5]={-10,5,15};
```

Бу ерда фақат массив бошидаги учта элементга бошланғич қийматлар берилган. Шуни айтиб ўтиш керакки, массивнинг бошидаги ёки ўртасидаги элементларига қийматлар бермасдан, унинг охиридаги элементларга бошланғич қиймат бериш мумкин эмас. Агарда массив элементларига бошланғич қиймат берилмаса, унда келишув бўйича *static* ва *extern* модификатори билан эълон қилинган массив учун элементларининг қиймати 0 сонига тенг деб, *auto* массивлар элементларининг бошланғич қийматлари номаълум ҳисобланади.

3) ўлчами кўрсатилмаган массив элементларини тўлиқ инициализациялаш:

```
int t[]={-10,5,15,4,3};
```

Бу мисолда массивни барча элементларига қийматлар берилган ҳисобланади, массив узунлиги компилятор томонидан бошланғич қийматлар сонига қараб аниқланади. Агарда массив узунлиги берил-маса, бошланғич қиймати берилиши шарт.

## 19-мавзу: Бир ўлчовли статик массивлар устида амаллар

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

- 1.Массивни киритиш
- 2.Энг катта ва энг кичик элементларни аниқлаш

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

Массивни эълон қилишга мисоллар:

```
char ch[4]={'a','b','c','d'}; // белгилар массиви  
int in[6]={10,20,30,40};    // бутун сонлар массиви
```

```

char str[]="abcd"; // сатр узунлиги 5 га тенг,
чунки
//унинг охирига '\0' белгиси
қўшилади
char str[]={'a','b','c','d'};
// юқоридаги сатрнинг бошқача
ёзилиши

```

**Масала.** Бир ой ичидаги кундалик ҳароратлар берилган. Ой учун ўртача ҳароратни ҳисоблаш программаси тузилсин.

Программа матни:

```

void main()
{
    const int n=30; // n=31; ёки n=29; ёки n=28;
    int temp[n];
    int i,s,temp_urtacha;
    cout<<"Kunlik haroratni kiriting:\n"
    for(i=0;i<n;i++)
    {
        cout << "\n temp["<<i<<"]=";
        cin >> temp[i];
    }
    for(i=0,s=0;i<n;i++) s+=temp[i];
    temp_urtacha=s/n;
    cout<<"Kunlik harorat :\n";
    for(i=0;i<n;i++)
        cout<<"\t temp["<<i<<"]="<<temp[i];
    cout<<"Oydagi o'rtacha harorat= "<<temp_urtacha;
    return;
}

```

## 20-мавзу: Кўп ўлчовли статик массивлар

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

**ДАРС Р Е Ж АСИ (асосий саволлар):**

- 1.С++ тилидаги программа тузилиши ва унинг компиляцияси
- 2.С++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

## Кўп ўлчамли статик массивлар

C++ тилида массивлар элементининг турига чекловлар қўйилмайди, лекин бу турлар чекли ўлчамдаги объектларнинг тури бўлиши керак. Чунки компилятор массивнинг хотирадан қанча жой (байт) эгаллашини ҳисоблай олиши керак. Хусусан, массив компонентаси массив бўлиши мумкин («*векторлар вектори*»), натижада *матрица* деб номланувчи икки ўлчамли массив ҳосил бўлади.

Агар матрицанинг элементи ҳам вектор бўлса, уч ўлчамли массивлар - *куб* ҳосил бўлади. Шу йўл билан ечилаётган масалага боғлиқ равишда ихтиёрий ўлчамдаги массивларни яратиш мумкин.

Икки ўлчамли массивнинг синтаксиси қуйидаги кўринишда бўлади:

<тур> <ном> [<узунлик >] [<узунлик>]

Масалан,  $10 \times 20$  ўлчамли ҳақиқий сонлар массивининг эълони:

```
float a[10][20];
```

Эълон қилинган *a* матрицани кўриниши 7.2-расмда келтирилган.

			$j$		
	$a_0$ :	$(a_{00},$	$a_{02}$	$\dots$	$\dots a_{018}, a_{019}),$
	$a_1$ :	$(a_{10},$	$a_{11},$	$\dots$	$\dots a_{118}, a_{119}),$
	$\dots$				
$i$	$a_i$ :	$(\dots,$	$\dots,$	$\dots a_{ij}$	$\dots \dots, \dots),$
	$\dots$				
	$a_9$ :	$(a_{90},$	$a_{91},$	$\dots$	$\dots a_{918}, a_{919}).$

7.2-расм. Икки ўлчамли массивнинг хотирадаги жойлашуви

Энди адрес нуқтаи - назаридан кўп ўлчамли массив элементларига мурожаат қилишни кўрайлик. Қуйидаги эълонлар берилган бўлсин:

```
int a[3][2];  
float b[2][2][2];
```

Биринчи эълонда икки ўлчамли массив, яъни 2 сатр ва 3 устундан иборат матрица эълон қилинган, иккинчисидан уч ўлчамли - 3 та  $2 \times 2$  матрицадан иборат бўлган массив эълон қилинган. Унинг элементларига мурожаат схемаси:

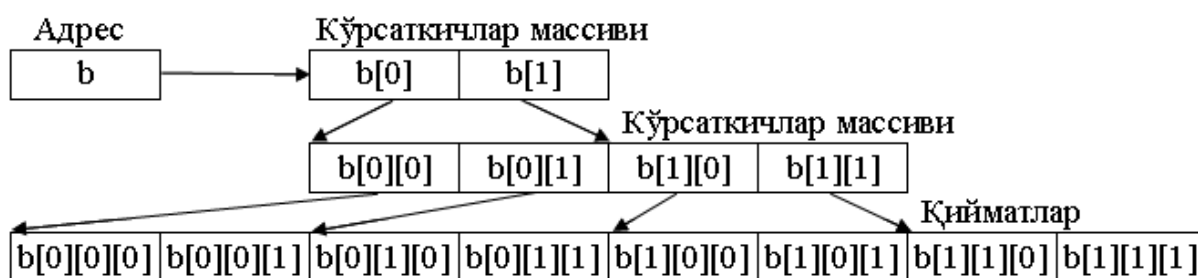


7.3-расм. Икки ўлчамли массив элементларига мурожаат

Бу ерда  $a[i]$  кўрсаткичда  $i$ -чи сатрнинг бошланғич адреси жойлашади, массив элементига  $a[i][j]$  кўринишидаги асосий мурожаатдан ташқари воситали мурожаат қилиш мумкин:  $*(*(a+i)+j)$  ёки  $*(a[i]+j)$ .

Массив элементларига мурожаат қилиш учун номдан кейин квадрат қавсда ҳар бир ўлчам учун индекс ёзилиши керак, масалан  $b[i][j][k]$ . Бу элементга воситали мурожаат ҳам қилиш мумкин ва унинг вариантлари:

$*(*(*(b+i)+j)+k)$  ёки  $*(*(b[i]+j)+k)$  ёки  $*(b[i][j]+k)$ .



7.3-расм. Уч ўлчамли массивнинг хотирада ташкил бўлиши

## 21-мавзу: Кўп ўлчовли статик массивлар устида амаллар

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

1. C++ тилидаги программа тузилиши ва унинг компиляцияси
2. C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

### Кўп ўлчамли массивларни инициализациялаш

Массивларни инициализациялаш қуйидаги мисолларда кўрсатилган:

```
int a[2][3]={0,1,2,10,11,12};
```

```
int b[3][3]={ {0,1,2},{10,11,12},{20,21,22}};
int c[3][3][3]={ { {0}}, { {100,101}, {110}},
    { {200,201,202}, {210,211,212}, {220,221,222}}};
```

Биринчи операторда бошланғич қийматлар кетма-кет ёзилган, иккинчи операторда қийматлар гуруҳлашган, учинчи операторда ҳам гуруҳлашган, лекин баъзи гуруҳларда охирги қийматлар берилмаган.

Мисол учун, матрицалар ва вектор кўпайтмасини -  $C = A \times b$  ҳисоблаш масаласини кўрайлик. Бу ерда  $A = \{a_{ij}\}$ ,  $b = \{b_j\}$ ,  $c = \{c_i\}$ ,

$0 \leq i < m, 0 \leq j < n$ . Ҳисоблаш формуласи -  $c_i = \sum_{j=0}^{n-1} a_{ij} b_j$ .

Программа матни:

```
void main()
{
    const int n=4,m=5;
    float a[m][n],b[n],c[m];
    int i,j;
    float s;
    for(i=0;i<m;i++)
        for(j=0;j<n;j++) cin>>a[i][j];
    for(i=0;i<m;i++) cin>>b[i];
    for(i=0;i<m;i++)
    {
        for (j=0,s=0;j<n;j++) s+=a[i][j]*b[j];
        c[i]=s;
    }
    for (i=0;i<m;i++) cout<<"\t c["<<i<<"]="<<c[i];
    return;
}
```

**22-мавзу:** Функциялар эълон қилиш ва аниқлаш. main() функцияси

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

ДАРС Р Е Ж А С И (асосий саволлар):

1. C++ тилидаги программа тузилиши ва унинг компиляцияси
2. C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

Программа таъминотини яратиш амалда мураккаб жараён ҳисобланади. Программа тузувчи программа комплексини бир бутун-ликда кўра билиши, ҳамда уни ташкил этувчи ҳар бир қисмнинг ички мазмунини англай олиши, уларни ўзаро фарқланиши ва боғланишларини ҳисобга олиши керак бўлади.

Программалашга тизимли ёндошув шундан иборатки, програм-ма тузувчи олдида қўйилган масала олдиндан иккита, учта ва ундан ортик нисбатан кичик масала остиларга бўлинади. Ўз навбатида бу масала остилари ҳам яна кичик масала остиларига бўлиниши мумкин. Бу жараён токи масалаларни (масала остиларни) оддий стандарт амаллар ёрдамида ечиш мумкин бўлгунча давом этади. Шу йўл билан масалани декомпозициялаш амалга оширилади.

Иккинчи томондан, программалашда шундай ҳолатлар кузатиладики, унда программанинг турли жойларида мазмунан бир хил алгоритмларни бажаришга тўғри келади. Алгоритмнинг бу бўлаклари асосий ечилаётган масаладан ажратиб олинган қандайдир масала остини ечишга мўлжалланган бўлиб, етарлича мустақил қийматга (натижага) эгадир. Мисол учун қуйидаги масалани кўрайлик:

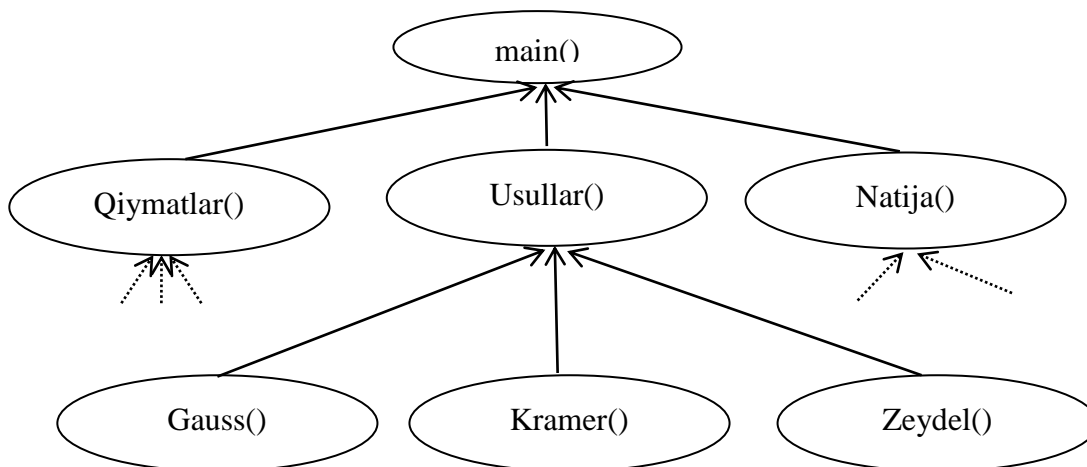
Берилган  $a_0, a_1, \dots, a_{30}, b_0, b_1, \dots, b_{30}, c_0, c_1, \dots, c_{30}$  ва  $x, y, z$  ҳақиқий сонлар учун

$$\frac{(a_0x^{30} + a_1x^{29} + \dots + a_{30})^2 - (b_0y^{30} + b_1y^{29} + \dots + b_{30})}{c_0(x+z)^{30} + c_1(x+z)^{29} + \dots + c_{30}}$$

ифоданинг қиймати ҳисоблансин.

Бу мисолни ечишда касрнинг сурат ва махражидаги ифодалар бир хил алгоритм билан ҳисобланади ва программада ҳар бир ифодани (масала ости) ҳисоблаш учун бу алгоритмни 3 марта ёзишга тўғри келади. Масаладаги 30-даражали кўпхадни ҳисоблаш алгоритмини, масалан, Горнер алгоритмини алоҳида, битта нусхада ёзиб, унга турли параметрлар - бир сафар  $a$  вектор ва  $x$  қийматини, иккинчи сафар  $b$  вектор ва  $y$  қийматини, ҳамда  $c$  вектор ва  $(x+z)$  қийматлари билан мурожаат қилиш орқали асосий масалани ечиш мумкин бўлади. Функциялар қўлланишининг яна бир сабабини қуйидаги масалада кўришимиз мумкин - берилган чизикли тенгламалар системасини Гаусс, Крамер, Зейдел усулларининг бирортаси билан ечиш талаб қилинсин. У ҳолда асосий программани қуйидаги бўлақларга бўлиш мақсадга мувофиқ бўлар эди: тенглама коэффицентларини ва озод ҳадларни киритиш бўлаги, ечиш усулини танлаш бўлаги, Гаусс, Крамер, Зейдел усулларини амалга ошириш учун алоҳида бўлақлар, натижани чоп қилиш бўлаги. Ҳар бир бўлақ учун ўз функциялар мажмуаси яратиб, зарур бўлганда уларга бош функция танасидан мурожаатни амалга ошириш орқали бош масала ечиш самарали ҳисобланади.

Бундай ҳолларда программани ихчам ва самарали қилиш учун C++ тилида программа бўлагини алоҳида ажратиб олиб, уни функция кўринишида аниқлаш имкони мавжуд (5.1 -расм).



5.1-расм. Масалани функциялар мажмуаси кўринишида ечиш

*Функция* - бу C++ тилида масала ечишнинг калит элементлари-дан биридир.

### Функция параметрлари ва аргументлари

Программада ишлатиладиган ҳар қандай функция эълон қилиниши керак. Одатда функциялар эълони сарлавҳа файлларда эълон қилинади ва *#include* директиваси ёрдамида программа матнига қўшилади.

Функция эълонини *функция прототипи* тавсифлайди (айрим ҳолларда *сигнатура* дейилади). Функция прототипи қуйидаги кўри-нишда бўлади:

<қайтарувчи қиймат тури> <функция номи>(<параметрлар рўйхати >);

Бу ерда <қайтарувчи қиймат тури> - функция ишлаши натижасида у томонидан қайтарадиган қийматнинг тури. Агар қайтариладиган қиймат тури кўрсатилмаган бўлса, келишув бўйича функция қайтара-диган қиймат тури *int* деб ҳисобланади, <параметрлар рўйхати>- вергул билан ажратилган функция параметрларининг тури ва номлари рўйхати. Параметр номини ёзмаса ҳам бўлади. Рўйхат бўш бўлиши ҳам мумкин. Функция прототипларига мисоллар:

```

int almashsin(int,int);
double max(double x,double y);
void func();
void chop_etish(void);

```

Функция прототипи тушириб қолдирилиши мумкин, агар прог-рамма матнида функция аниқланиши уни чақирадиган функциялар матнидан олдин ёзилган бўлса. Лекин бу ҳолат яхши услуб ҳисоб-ланмайди, айниқса



ўзаро бир - бирига мурожаат қилувчи функциялар-ни эълон қилишда муаммолар юзага келиши мумкин.

*Функция аниқланиши* - функция сарлавҳаси ва фигурали қавсга ('{'','}') олинган қандайдир амалий мазмунга эга танадан иборат бўлади. Агар функция қайтарувчи тури *void* туридан фарқли бўлса, унинг танасида албатта мос турдаги параметрга эга *return* оператори бўлиши шарт. Функция танасида биттадан ортиқ *return* оператори бўлиши мумкин. Уларнинг ихтиёрий бирортасини бажариш орқали функциядан чиқиб кетилади. Агар функция қайтарадиган қиймат уни чақирган функцияда ишлатилмайдиган бўлса, функциядан чиқиш учун параметрсиз *return* оператори ишлатилиши мумкин ёки умуман *return* ишлатилмайди. Охирги ҳолда функциядан чиқиш - охирги ёпилувчи қавсга етиб келганда рўй беради.

Функция программанинг бирорта модулида ягона равишда аниқланиши керак, унинг эълони эса функцияни ишлатадиган модул-ларда бир неча марта ёзилиши мумкин. Функция аниқланишида сарлавҳадаги барча параметрлар номлари ёзилиши шарт.

Одатда программада функция маълум бир ишни амалга ошириш учун чақирилади. Функцияга мурожаат қилганда, у қўйилган масала-ни ечади ва ўз ишини тугатишида қандайдир қийматни натижа сифатида қайтаради.

*Функцияни чақириши* учун унинг номи ва ундан кейин қавс ичида аргументлар рўйхати берилади:

<функция номи>(<аргумент<sub>1</sub>>, <аргумент<sub>2</sub>>, ..., <аргумент<sub>n</sub>>);

Бу ерда ҳар бир <аргумент<sub>i</sub>> - функция танасига узатиладиган ва кейинчалик ҳисоблаш жараёнида ишлатиладиган ўзгарувчи, ифода ёки ўзгармасдир. Аргументлар рўйхати бўш бўлиши мумкин.

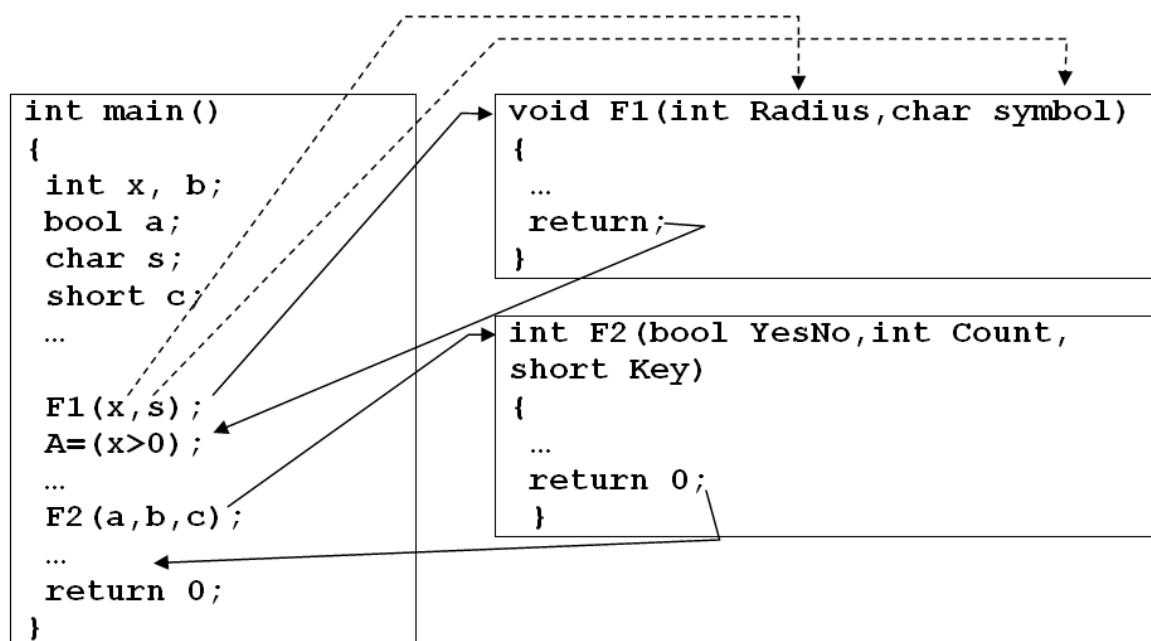
Функциялар ҳам ўз танасида бошқа функцияларни, ўзини ҳам чақириши мумкин. Ўз танасида ўзини чақирадиган функцияларга *рекурсив функциялар* дейилади.

Олдинги бобларда таъкидлаб ўтилганидек, C++ тилидаги ҳар қандай программада албатта *main()* бош функцияси бўлиши керак. Айни шу функцияни юклагич томонидан чақирилиши билан програм-ма бажарилиши бошланади.

5.1- расмда бош функциядан бошқа функцияларни чақириш ва улардан қайтиш схемаси кўрсатилган.

Программа *main()* функциясини бажаришдан бошланади ва «*f1(x,y);*» - функция чақиришгача давом этади ва кейинчалик бошқа-рув *f1(x,y)* функция танасидаги амалларни бажаришга ўтади. Бунда *Radius* параметрининг қиймати сифатида функция *x* ўзгарувчи қийматини, *symbol* параметри сифатида *y* ўзгарувчисининг қиймати ишлатилади. Функция танаси *return* операторигача бажарилади. *return* оператори бошқарувни *main()* функцияси танасидаги *f1()* функцияси чақирилган оператордан кейинги операторга ўтишни таъминлайди, яъни функциядан қайтиш рўй

беради. Шундан кейин *main()* функция-си операторлари бажарилишда давом этади ва «*f2(a,b,c);*» - функция чақириши орқали бошқарув *f2()* функция танасига ўтади ва ҳисоблаш жараёнида мос равишда *YesNo* сифатида *a* ўзгарувчисининг, *count* сифатида *b* ўзгарувчисининг ва *key* сифатида *c* ўзгарувчисининг қийматлари ишлатилади. Функция танасидаги *return* оператори ёки охириги оператор бажаргандан кейин автоматик равишда бош функцияга қайтиш амалга оширилади.



5.1-расм. Бош функциядан бошқа функцияларни чақириш ва қайтиш

Аксарият ҳолларда *main()* функциясининг параметрлар рўйхати бўш бўлади. Агар юкланувчи программани ишга туширишда, буйруқ сатри орқали юкланувчи программа ишга туширилганда, унга параметрларни узатиш (бериш) зарур бўлса, *main()* программаси функциясининг синтаксиси ўзгаради:

```
int main(int argc, char* argv[]);
```

Бу ерда *argc* - узатиладиган параметрлар сони, *argv[]*- бир-биридан пунктуация белгилари (ва пробел) билан ажратилган параметрлар рўйхатини ўз ичига олган массивга кўрсаткич.

Қуйида функцияларни эълон қилиш, чақириш ва аниқлашга мисоллар келтирилган:

```
// функциялар эълони
int Mening_funksiyam(int Number, float Point);
char Belgini_uqish();
void bitni_urnatish(short Num);
void Amal_yoq(int, char);

// функцияларни чақириш
result=Mening_funksiyam(Varb1, 3.14);
```

```

symb=Belgini_uqish();
bitni_urnatish(3);
Amal_yoq(2,Smb1);

// функцияларни аниқлаш

int Mening_funksiyam(int Number,float Point);
{
    int x;
    //...
    return x;
}
char Belgini_uqish()
{
    char Symbol;
    cin>>Symbol;
    return Symbol;
};
void bitni_urnatish(short number)
{
    global_bit=global_bit | number;
};
void Amal_yoq(int x,char ch){};

```

Функциянинг программадаги ўрнини янада тушунарли бўлиши учун сон квадрaтини ҳисоблаш масаласида функциядан фойдаланишни кўрайлик.

Функция прототипини “sarlavha.h” сарлавҳа файлида жойлаштирамиз:

```

long Son_Kvadrati(int);

```

Асосий программага ушбу сарлавҳа файлини қўшиш орқали *Son\_Kvadrati()* функция эълони программа матнига киритилади:

```

#include <iostream.h>
#include "sarlavha.h"
int main()
{
    int Uzgaruvchi=5;
    cout<<Son_Kvadrati(Uzgaruvchi);
    return 0;
}
long Son_Kvadrati(int x) {return x*x;}

```

Худди шу масалани сарлавҳа файлидан фойдаланмаган ҳолда, функция эълонини программа матнига ёзиш орқали ҳам ҳал қилиш мумкин:

```

#include <iostream.h>

```

```

long Son_Kvadrati(int);
int main()
{
    int Uzgaruvchi=5;
    cout<<Son_Kvadrati(Uzgaruvchi);
    return 0;
}
long Son_Kvadrati(int x){return x*x;}

```

Программа ишлашида ўзгариш бўлмади ва натижа сифатида экранга 25 сонини чоп этади.

**Масала.** Иккита туб сон «эгизак» дейилади, агар улар бир-биридан 2 фарқ қилса (масалан, 41 ва 43 сонлари). Берилган натурал  $n$  учун  $[n..2n]$  кесмадаги барча «эгизак» сонлар жуфтликлари чоп этилсин. Масалани ечиш учун берилган  $k$  сонини туб сон ёки йўқлиги аниқловчи мантиқий функцияни тузиш зарур бўлади. Функцияда  $k$  сони  $2..k/2$  гача сонларга бўлинади, агар  $k$  бу сонларнинг бирортасига ҳам бўлинмаса, у туб сон ҳисобланади ва функция *true* қийматини қайтаради. Бош функцияда, берилган  $n$  учун  $[n..2n]$  ораликдаги  $(n, n+2), (n+1, n+3), \dots, (2n-2, 2n)$  сон жуфтликларини туб сонлар эканлиги текширилади ва шартни қаноатлантирган жуфтликлар чоп этилади.

Программа матни:

```

bool TubSon(unsigned long k);
int main()
{
    unsigned long n,i;
    unsigned char egizak=0;
    cout<<"n -> ";
    cin>>n;
    cout<<' ['<<n<<".."<<2*n<<' ]' ;
    for(i=n; i<=2*n-2; i++)
        if(TubSon(i) && TubSon(i+2))
        {
            if (!egizak)
                cout<<" oraliq'idagi egizak tub sonlar:\n";
            else cout<<" ";
            egizak=1;
            cout<<' {'<<i<<' , '<<i+2<<' }' ;
        };
    if(!egizak)
        cout<<" oraliq'ida egizak tub sonlar yo'q.";
    else cout<<' .' ;
    return 0;
}
bool TubSon(unsigned long k)

```

```

{
    unsigned long m;
    for (m=2; m<=k/2; m++)
        if (k%m==0) return false;
    return true;
}

```

Натурал  $n$  сони учун 100 киритилса, программа қуйидаги сонлар жуфтликларини чоп қилади:

```

[100..200] oraliq'idagi egizak tub sonlar:
{101,103}; {107,109}; {137,139}; {149,151};
{179,181}; {191,193}; {197,199}.

```

### Келишув бўйича аргументлар

C++ тилида функция чақирилганда айрим аргументларни тушириб қолдириш мумкин. Бунга функция прототида ушбу пара-метрларни келишув бўйича қийматини кўрсатиш орқали эришиш мумкин. Масалан, қуйида прототиби келтирилган функция турли чақиришга эга бўлиши мумкин:

```

//функция прототиби
void Butun_Son(int I,bool Bayroq=true,char
Blg='\n') ;
//функцияни чақириш вариантлари
Butun_Son(1,false,'a') ;
Butun_Son(2,false) ;
Butun_Son(3) ;

```

Биринчи чақирувда барча параметрлар мос аргументлар орқали қийматларини қабул қилади, иккинчи ҳолда  $I$  параметри 2 қийматини, *bayroq* параметри *false* қийматини ва *Blg* ўзгарувчиси келишув бўйича '\n' қийматини қабул қилади.

Келишув бўйича қиймат беришнинг битта шarti бор - пара-метрлар рўйхатида келишув бўйича қиймат берилган параметрлардан кейинги параметрлар ҳам келишув бўйича қийматга эга бўлишлари шарт. Юқоридаги мисолда  $I$  параметри келишув бўйича қиймат қабул қилинган ҳолда, *Bayroq* ёки *Blg* параметрлари қийматсиз бўлиши мумкин эмас. Мисол тариқасида берилган сонни кўрсатилган аниқликда чоп этувчи программани кўрайлик. Қўйилган масалани ечишда сонни даражага ошириш функцияси - *pow()* ва сузувчи нуқтали узун сондан модул олиш *fabsl()* функциясидан фойдаланилади. Бу функциялар прототиби «math.h» сарлавҳа файлида жойлашган (3-илова қаранг):

```

#include <iostream.h>
#include <math.h>
void Chop_qilish(double Numb,double Aniqlik=1,
                bool Bayroq=true);

```

```

int main()
{
    double Mpi=-3.141592654;
    Chop_qilish(Mpi,4,false);
    Chop_qilish(Mpi,2);
    Chop_qilish(Mpi);
    return 0;
}
void Chop_qilish(double Numb,double Aniqlik=1,
                 bool Bayroq=true)
{
    if(!Bayroq) Numb=fabs1(Numb);
    Numb=(int)(Numb*pow(10,Aniqlik));
    Numb=Numb/pow(10,Aniqlik);
    cout<<Numb<<' \n' ;
}

```

Программада сонни турли аниқликда (*Aniqlik* параметри қиймати орқали) чоп этиш учун ҳар хил вариантларда *Chop\_qilish()* функцияси чақирилган. Программа ишлаши натижасида экранда куйидаги сонлар чоп этилади:

```

3.1415
-3.14
-3.1

```

Параметрнинг келишув бўйича бериладиган қиймати ўзгармас, ўзгарувчи ёки қандайдир функция томонидан қайтарадиган қиймат бўлиши мумкин.

## 23-мавзу: Локал ва глобал ўзгарувчилар

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

ДАРС Р Е Ж А С И (асосий саволлар):

1. C++ тилидаги программа тузилиши ва унинг компиляцияси
2. C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

### Кўриниш соҳаси. Локал ва глобал ўзгарувчилар

Ўзгарувчилар функция танасида ёки ундан ташқарида эълон қилиниши мумкин. Функция ичида эълон қилинган ўзгарувчиларга *локал*

*ўзгарувчилар* дейилади. Бундай ўзгарувчилар хотирадаги программа стекида жойлашади ва фақат ўзи эълон қилинган функция танасида амал қилади. Бошқарув асосий функцияга қайтиши билан локал ўзгарувчилар учун ажратилган хотира бўшатилади (ўчирилади).

Ҳар бир ўзгарувчи ўзининг амал қилиш соҳаси ва яшаш вақти хусусиятлари билан характерланади.

Ўзгарувчи *амал қилиш соҳаси* деганда ўзгарувчини ишлатиш мумкин бўлган программа соҳаси (қисми) тушунилади. Бу тушунча билан ўзгарувчининг *кўриниш соҳаси* узвий боғланган. Ўзгарувчи амал қилиш соҳасидан чиққанда кўринмай қолади. Иккинчи томон-дан, ўзгарувчи амал қилиш соҳасида бўлиши, лекин кўринмаслиги мумкин. Бунда кўриниш соҳасига рухсат бериш амали “::” ёрдамида кўринмас ўзгарувчига мурожат қилиш мумкин бўлади.

Ўзгарувчининг *яшаш вақти* деб, у мавжуд бўлган программа бўлагининг бажарилишига кетган вақт интервалига айтилади.

Локал ўзгарувчилар ўзлари эълон қилинган функция ёки блок чегарасида кўриниш соҳасига эга. Блокдаги ички блокларда худди шу номдаги ўзгарувчи эълон қилинган бўлса, ички блокларда бу локал ўзгарувчи ҳам амал қилмай қолади. Локал ўзгарувчи яшаш вақти - блок ёки функцияни бажариш вақти билан аниқланади. Бу ҳол шуни англатадики, турли функцияларда бир-бирига умуман боғлиқ бўлмаган бир хил номдаги локал ўзгарувчиларни ишлатиш мумкин.

Қуйидаги программада *main()* ва *sum()* функцияларида бир хил номдаги ўзгарувчиларни ишлатиш кўрсатилган. Програмада иккита соннинг йиғиндиси ҳисобланади ва чоп этилади:

```
#include <iostream.h>
// функция прототипи
int sum(int a,int b);
int main()
{
    // локал ўзгарувчилар
    int x=2;
    int y=4;
    cout<<sum(x,y);
    return 0;
}
int sum(int a,int b)
{
    // локал ўзгарувчи
    int x=a+b;
    return x;
}
```

Глобал ўзгарувчилар программа матнида функция аниқланиши-дан ташқарида эълон қилинади ва эълон қилинган жойидан бошлаб программа охиригача амал қилади.

```
#include <iostream.h>
int f1(); int f2();
int main()
{
    cout<<f1()<<" "<<f2()<<endl;
    return 0;
}
int f1()
{
    return x;// компиляция хатоси рўй беради
}
int x=10; // глобал ўзгарувчи эълони
int f2(){return x*x;}
```

Юқорида келтирилган программада компиляция хатоси рўй беради, чунки *f1()* функция учун *x* ўзгарувчиси номаълум ҳисоб-ланади.

Программа матнида глобал ўзгарувчиларни улар эълонидан кейин ёзилган ихтиёрий функцияда ишлатиш мумкин. Шу сабабли, глобал ўзгарувчилар программа матнининг бошида ёзилади. Функция ичидан глобал ўзгарувчига мурожат қилиш учун функцияда унинг номи билан мос тушадиган локал ўзгарувчилар бўлмаслиги керак. Агар глобал ўзгарувчи эълонида унга бошланғич қиймат берилмаган бўлса, уларнинг қиймати 0 ҳисобланади. Глобал ўзгарувчининг амал қилиш соҳаси унинг кўриниш соҳаси билан устма-уст тушади.

Шуни қайд этиш керакки, тажрибали программа тузувчилар имкон қадар глобал ўзгарувчиларни ишлатмасликка ҳаракат қилиша-ди, чунки бундай ўзгарувчилар қийматини программанинг ихтиёрий жойидан ўзгартириш хавфи мавжудлиги сабабли программа ишлаши-да мазмунан хатолар юзага келиши мумкин. Бу фикримизни тасдиқ-ловчи программани кўрайлик.

```
# include <iostream.h>
// глобал ўзгарувчи эълони
int test=100;
void Chop_qilish(void);
int main()
{
    //локал ўзгарувчи эълони
    int test=10;
    //глобал ўзгарувчи чоп қилиш функциясини чақириш
    Chop_qilish();
    cout<<"Lokal o'zgaruvchi: "<<test<<' \n' ;
    return 0;
```



```

}
void Chop_qilish(void)
{
    cout<<"Global o'zgaruvchi: "<<test<<' \n' ;
}

```

Программа бошида *test* глобал ўзгарувчиси 100 қиймати билан эълон қилинади. Кейинчалик, *main()* функциясида *test* номи билан локал ўзгарувчиси 10 қиймати билан эълон қилинади. Программада, *Chop\_qilish()* функциясига мурожаат қилинганида, асосий функция танасидан вақтинча чиқилади ва натижада *main()* функциясида эълон қилинган барча локал ўзгарувчиларга мурожаат қилиш мумкин бўлмай қолади. Шу сабабли *Chop\_qilish()* функциясида глобал *test* ўзгарувчисининг қийматини чоп этилади. Асосий программага қайтилгандан кейин, *main()* функциясидаги локал *test* ўзгарувчиси глобал *test* ўзгарувчисини «*беркутади*» ва локал *test* ўзгарувчини қиймати чоп этилади. Программа ишлаши натижасида экранга қуйидаги натижалар чоп этилади:

```

Глобал ўзгарувчи: 100
Локал ўзгарувчи: 10

```

#### **:: амали**

Юқорида қайд қилингандек, локал ўзгарувчи эълони худди шу номдаги глобал ўзгарувчини «*беркутади*» ва бу жойдан глобал ўзгарувчига мурожат қилиш имкони бўлмай қолади. C++ тилида бундай ҳолатларда ҳам глобал ўзгарувчига мурожат қилиш имко-нияти сақланиб қолинган. Бунинг учун «*кўриниш соҳасига рухсат бериш*» амалидан фойдаланиш мумкин. Мисол тариқасида қуйидаги програмани келтирамыз:

```

#include <iostream.h >
int uzg=5; //глобал ўзгарувчи эълони
int main()
{
    int uzg=70; //локал ўзгарувчи эълони
    cout<<uzg<<' \n' ; // локал ўзгарувчини чоп
этиш
    cout<<::uzg <<' \n' ; // глобал ўзгарувчини чоп
этиш
    return 0;
}

```

Программа ишлаши натижасида экранга олдин 70 ва кейин 5 сонлари чоп этилади.

## Хотира синфлари

Ўзгарувчиларнинг кўриниш соҳаси ва амал қилиш вақтини аниқловчи ўзгарувчи модификаторлари мавжуд (5.1-жадвал).

5.1-жадвал. Ўзгарувчи модификаторлари

Модификатор	Қўлланиши	Амал қилиш соҳаси	Яшаш даври
auto	локал	блок	вақтинча
register	локал	блок	вақтинча
extern	глобал	блок	вақтинча
static	локал	блок	доимий
	глобал	файл	доимий
volatile	глобал	файл	доимий

**Автомат ўзгарувчилар.** *auto* модификатори локал ўзгарувчилар эълонида ишлатилади. Одатда локал ўзгарувчилар эълонида бу модификатор келишув бўйича қўлланилади ва шу сабабли амалда уни ёзишмайди:

```
#include <iostream.h>
int main()
{
    auto int X=2; //бу эълон "int X=2;" билан
    эквивалент
    cout<<X;
    return 0;
}
```

*auto* модификатори блок ичида эълон қилинган локал ўзгарувчиларга қўлланилади. Бу ўзгарувчилар блокдан чиқиши билан автоматик равишда йўқ бўлиб кетади.

**Регистр ўзгарувчилар.** *register* модификатори компиляторга, кўрсатилган ўзгарувчини процессор регистрларига жойлаштиришга ҳаракат қилишни тайинлайди. Агар бу ҳаракат натижа бермаса ўзгарувчи *auto* туридаги локал ўзгарувчи сифатида амал қилади.

Ўзгарувчиларни регистрларда жойлаштириш программа кодини бажариш тезлиги бўйича оптималлаштиради, чунки процессор хотирадаги берилганларга нисбатан регистрдаги қийматлар билан анча тез ишлайди. Лекин регистрлар сони чекланганлиги учун ҳар доим ҳам ўзгарувчиларни регистрларда жойлаштиришнинг иложи бўлмайди.

```
#include < iostream.h >
int main()
{
    register int Reg;
    ...
    return 0;
}
```

```
}
```

*register* модификатори фақат локал ўзгарувчиларига нисбатан қўлланилади, глобал ўзгарувчиларга қўллаш компиляция хатосига олиб келади.

**Ташқи ўзгарувчилар.** Агар программа бир нечта модулдан иборат бўлса, улар қандайдир ўзгарувчи орқали ўзаро қиймат алма-шишлари мумкин (файллар орасида). Бунинг учун ўзгарувчи бирорта модулда глобал тарзда эълон қилинади ва у бошқа файлда (модулда) кўриниши учун у ерда *extern* модификатори билан эълон қилиниши керак бўлади. *extern* модификатори ўзгарувчини бошқа файлда эълон қилинганлигини билдиради. Ташқи ўзгарувчилар ишлатилган прог-рамmani кўрайлик.

```
//Sarlavha.h файлида
void Bayroq_Almashsin(void);

// modul_1.cpp файлида
bool Bayroq;
void Bayroq_Almashsin(void){Bayroq=!Bayroq;}

// masala.cpp файлида
#include <iostream.h>
#include <Sarlavha.h>
#include <modul_1.cpp>
extern bool Bayroq;
int main()
{
    Bayroq_Almashsin();
    if(Bayroq)
        cout<<"Bayroq TRUE"<<endl;
    else cout<<"Bayroq FALSE"<<endl;
    return 0;
}
```

Олдин “sarlavha.h” файлида *Bayroq\_Almashsin()* функция сарлав-ҳаси эълон қилинади, кейин “modul\_1.cpp” файлида ташқи ўзгарувчи эълон қилинади ва *Bayroq\_Almashsin()* функциясининг танаси аниқла-нади ва ниҳоят, “masala.cpp” файлида *Bayroq* ўзгарувчиси ташқи деб эълон қилинади.

**Статик ўзгарувчилар.** Статик ўзгарувчилар *static* модифика-тори билан эълон қилинади ва ўз хусусиятига кўра глобал ўзгарувчи-ларга ўхшайди. Агар бу турдаги ўзгарувчи глобал бўлса, унинг амал қилиш соҳаси - эълон қилинган жойдан программа матнининг охири-гача бўлади. Агар статик ўзгарувчи функция ёки блок ичида эълон қилинадиган бўлса, у функция ёки блокка биринчи киришда иници-ализация қилинади. Ўзгарувчининг бу қиймати функция кейинги чақирилганида ёки блокка қайта киришда сақланиб қолади ва бу қийматни ўзгартириш мумкин. Статик ўзгарувчиларни ташқи деб эълон қилиб бўлмайди.

Агар статик ўзгарувчи инициализация қилинмаган бўлса, унинг биринчи мурожатдаги қиймати 0 ҳисобланади.

Мисол тариқасида бирорта функцияни неча маротаба чақирилганлигини аниқлаш масаласини кўрайлик:

```
#include <iostream.h >
int Sanagich(void);
const int n=30;
int main()
{
    int natija;
    for(int i=0;i<n;i++)
        natija=Sanagich();
    cout<<natija;
    return 0;
}
int Sanagich(void)
{
    static short sanagich=0;
    ...
    sanagich++;
    return sanagich;
}
```

Бу ерда асосий функциядан counter статик ўзгарувчига эга *Sanagicht()* функцияси 30 марта чақирилади. Функция биринчи марта чақирилганда *sanagich* ўзгарувчига 0 қийматини қабул қилади ва унинг қиймати бирга ортган ҳолда функция қиймати сифатида қайтарилади. Статик ўзгарувчилар қийматларини функцияни бир чақирилишидан иккинчисига сақланиб қолиниши сабабли, кейинги ҳар бир чақиришларда *sanagich* қиймати биттага ортиб боради.

**Масала.** Берилган ишорасиз бутун  $n$  соннинг барча туб бўлувчилари аниқлансин. Масалани ечиш алгоритми куйидаги такрорла-нувчи жараёндан иборат бўлади: берилган сон туб сонга (1-қадамда 2 га) бўлинади. Агар қолдиқ 0 бўлса, туб сон чоп қилинади ва бўлинувчи сифатида бўлинма олинади яна шу туб сонга бўлинади, акс ҳолда навбатдаги туб сон олинади. Такрорлаш навбатдаги туб сон бўлинувчига тенг бўлгунча давом этади.

Программа матни:

```
#include<iostream.h>
#include<math.h>
int Navb_tub();
int main()
{
    unsigned int n,p;
    cout<<"\nn qiymatini kiritng: ";
```

```

cin>>n;
cout<<"\n1";
p=Navb_tub();
while (n>=p)
{
    if (n%p==0)
    {
        cout<<' '*<<p;
        n=n/p;
    }
    else p=Navb_tub();
}
return 0;
}
int Navb_tub()
{
    static unsigned int tub=1;
    for(;;)
    {
        tub++;
        short int ha_tub=1;
        for(int i=2;i<=tub/2;i++)
            if(tub%i==0)ha_tub=0;
        if(ha_tub)return tub;
    }
    return 0;
}

```

Программада навбатдаги туб сонни ҳосил қилиш функция кўри-нишида амалга оширилган. *Navb\_tub()* функциясининг ҳар чақирили-шида охирги туб сондан кейинги туб сон топилади. Охирги туб сонни «эслаб» қолиш учун *tub* ўзгарувчиси *static* қилиб аниқланган.

Программа ишга тушганда клавиатурадан *n* ўзгарувчисининг қиймати сифатида 60 сони киритилса, экранга қуйидаги кўпайтма чоп этилади:

**1\*2\*2\*3\*5**

**volatile синфи ўзгарувчилари.** Агар программада ўзгарувчини бирорта ташқи қурилма ёки бошқа программа билан боғлаш учун ишлатиш зарур бўладиган бўлса, у *volatile* модификатори билан эълон қилинади. Компилятор бундай модификаторли ўзгарувчини регистрга жойлаштиришга ҳаракат қилмайди. Бундай ўзгарувчилар эълонига мисол қуйида келтирилган:

```

volatile short port_1;
volatile const int Adress=0x00A2;

```

Мисолдан кўришиб турибдики, *volatile* модификаторли ўзгармас ҳам эълон қилиниши мумкин.

## 24-мавзу: Функция параметрларида статик массивлардан фойдаланиш

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

ДАРС Р Е Ж А С И (асосий саволлар):

1. C++ тилидаги программа тузилиши ва унинг компиляцияси
2. C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

Функциялар массивни параметр сифатида ишлатиши ва уни функциянинг натижаси сифатида қайтариши мумкин.

Агар массив параметр орқали функцияга узатилса, элементлар сонини аниқлаш муаммоси туғилади, чунки массив номидан унинг узунлигини аниқлашнинг иложи йўқ. Айрим ҳолларда, масалан, белгилар массиви сифатида аниқланган сатр (ASCIIZ сатрлар) билан ишлаганда массив узунлигини аниқлаш мумкин, чунки сатрлар ‘\0’ белгиси билан тугайди.

Мисол учун:

```
#include <iostream.h>
int satr_uzunligi(char s[])//массив параметр сифатида
{
    int m=0;
    while(s[m++]);
    return m-1;
}
void main()
{
    char z[]="Ushbu satr uzunligi = ";
    cout<<z<<satr_uzunligi(z);
}
```

Функция параметри сатр бўлмаган ҳолларда фиксирланган узунликдаги массивлар ишлатилади. Агар турли узунликдаги массивларни узатиш зарур бўлса, массив ўлчамларини параметр сифатида узатиш мумкин ёки бу мақсадда глобал ўзгарувчидан фойдаланишга тўғри келади.

Мисол:

```
#include <iostream.h>
float yigindi(int n,float *x) //бу иккинчи усул
{
    float s=0;
    for(int i=0;i<n;i++)s+=x[i];
    return s;
}
void main()
{
    float E[]={1.2,2.0,3.0,4.5,-4.0};
    cout<<yigindi(5,E);
}
```

Массив номи кўрсаткич бўлганлиги сабабли массив элементларининг қийматлари функция ичида ўзгартирилса, бу ўзгаришлар функциядан чиққандан кейин ҳам сақланиб қолади.

```
#include <iostream.h>
void vector_01(int n,int*x,int*y) //бу иккинчи усул
{
    for(int i=0;i<n;i++)
        y[i]=x[i]>0?1:0;
}
void main()
{
    int a[]={1,2,-4,3,-5,0,4};
    int c[7];
    vector_01(7,a,c);
    for(int i=0;i<7;i++) cout<<'\\t'<<c[i]; }
}
```

**Масала.** Бутун турдаги ва элементлари камаймайдиган ҳолда тартибланган бир ўлчамли иккита массивларни ягона массивга, тартиб сақланган ҳолда бирлаштирилсин.

Программа матни:

```
#include <iostream.h>
\\бутун турдаги массивга кўрсаткич қайтарадиган
\\функция
int * massiv_ulash(int,int*,int,int*);
void main()
{
    int c[]={-1,2,5,9,10},d[]={1,7,8};
    int * h;
    h=massiv_ulash(5,c,3,d);
    for(int i=0;i<8;i++)cout<<'\\t'<<h[i];
    delete[]h;
}
int * massiv_ulash(int n,int *a ,int m,int *b);
{
```

```

int * x=new int[n+m];
int ia=0,ib=0,ix=0;
while (ia<n && ib<m)
    a[ia]>b[ib]?x[ix++]=b[ib++]:x[ix++]=a[ia++];
while(ib<m)x[ix++]=b[ib++];
while(ia<n)x[ix++]=a[ia++];
return x;
}

```

Программа ишлаши натижасида экранга

-1    1        2        5        7        8        9        10

сонлар кетма-кетлиги чоп этилади.

## 25-мавзу: Рекурсив функциялар

**Ажратилган соат:**            2 соат  
**Машғулот тури:**            маъруза

ДАРС Р Е Ж А С И (асосий саволлар):

- 1.С++ тилидаги программа тузилиши ва унинг компиляцияси
- 2.С++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оким, компилятор.

### Рекурсив функциялар

Юқорида қайд қилингандек *рекурсия* деб функция танасида шу функциянинг ўзини чақиришига айтилади. Рекурсия икки хил бўлади:

- 1) *оддий* - агар функция ўз танасида ўзини чақирса;
- 2) *воситали* - агар биринчи функция иккинчи функцияни чақирса, иккинчиси эса ўз навбатида биринчи функцияни чақирса.

Одатда рекурсия математикада кенг қўлланилади. Чунки аксарият математик формулалар рекурсив аниқланади. Мисол тариқасида факториални ҳисоблаш формуласини

$$n! = \begin{cases} 1, & \text{агар } n = 0; \\ n * (n-1)!, & \text{агар } n > 0, \end{cases}$$

ва соннинг бутун даражасини ҳисоблашни кўришимиз мумкин:

$$x^n = \begin{cases} 1, & \text{агар } n = 0; \\ x * x^{n-1}, & \text{агар } n > 0. \end{cases}$$



Кўришиб турибдики, навбатдаги қийматни ҳисоблаш учун функциянинг «олдинги қиймати» маълум бўлиши керак. C++ тилида рекурсия математикадаги рекурсияга ўхшаш. Буни юқоридаги мисоллар учун тузилган функцияларда кўриш мумкин. Факториал учун:

```
long int F(int n)
{
    if(!n) return 1;
    else return n*F(n-1);
}
```

Берилган ҳақиқий  $x$  сонинг  $n$ - даражасини ҳисоблаш функцияси:

```
double Butun_Daraja(double x, int n)
{
    if(!n) return 1;
    else return x*Butun_Daraja(x,n-1);
}
```

Агар факториал функциясига  $n>0$  қиймат берилса, куйидаги ҳолат рўй беради: шарт операторининг *else* шохдаги қиймати ( $n$  қиймати) стекда эслаб қолинади. Ҳозирча қиймати номаълум  $n-1$  факториални ҳисоблаш учун шу функциянинг ўзи  $n-1$  қиймати билан билан чақирилади. Ўз навбатида, бу қиймат ҳам эслаб қолинади (стекка жойланади) ва яна функция чақирилади ва ҳакоза. Функция  $n=0$  қиймат билан чақирилганда *if* операторининг шарти ( $!n$ ) рост бўлади ва «*return 1;*» амали бажарилиб, айти шу чақириш бўйича 1 қиймати қайтарилади. Шундан кейин «тескари» жараён бошланади - стекда сақланган қийматлар кетма-кет олинадилар ва кўпайтирилади: охириги қиймат - аниқлангандан кейин (1), ундан олдинги сақланган қийматга 1 қийматига кўпайтириб  $F(1)$  қиймати ҳисобланади, бу қиймат 2 қийматига кўпайтириш билан  $F(2)$  топилади ва ҳакоза. Жараён  $F(n)$  қийматини ҳисоблашгача «кўтарилиб» боради. Бу жараённи,  $n=4$  учун факториал ҳисоблаш схемасини 5.2-расмда кўриш мумкин:

↓	$F(4)=4 \cdot F(3)$	↓	$F(4)=4 \cdot F(3)$	↓	$F(4)=4 \cdot F(3)$	↓	$F(4)=4 \cdot F(3)$	↑	$F(4)=4 \cdot 6$
↓	$F(3)=3 \cdot F(2)$	↓	$F(3)=3 \cdot F(2)$	↓	$F(3)=3 \cdot F(2)$	↑	$F(3)=3 \cdot 2$		
↓	$F(2)=2 \cdot F(1)$	↓	$F(2)=2 \cdot F(1)$	↑	$F(2)=2 \cdot 1$				
↓	$F(1)=1 \cdot F(0)$	↑	$F(1)=1 \cdot 1$						
↑	$F(0)=1$								

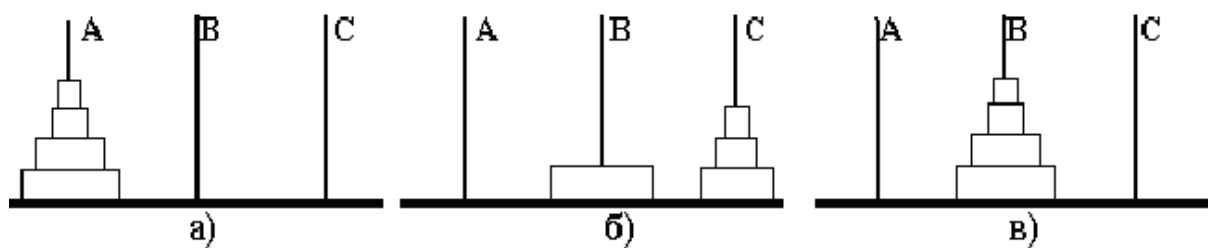
5.2-расм.  $4!$  ҳисоблаш схемаси

Рекурсив функцияларни тўғри амал қилиши учун рекурсив чақиришларнинг тўхташ шарти бўлиши керак. Акс ҳолда рекурсия тўхтамаслиги ва ўз навбатида функция иши тугамаслиги мумкин. Факториал ҳисоблашида рекурсив тушишларнинг тўхташ шарти функция параметри  $n=0$  бўлишидир (шарт операторининг рост шох).

Ҳар бир рекурсив мурожаат қўшимча хотира талаб қилади - функцияларнинг локал объектлари (ўзгарувчилари) учун ҳар бир мурожаатда стекдан янгидан жой ажратилади. Масалан, рекурсив функцияга 100 марта мурожаат бўлса, жами 100 локал объектларнинг мажмуаси учун жой ажратилади. Айрим ҳолларда, яъни рекурсиялар сони етарлича катта бўлганда, стек ўлчами чекланганлиги сабабли (реал режимда 64Кб ўлчамгача) у тўлиб кетиши мумкин. Бу ҳолатда программа ўз ишини «Стек тўлиб кетди» хабари билан тўхтади.

Қуйида, рекурсия билан самарали ечиладиган «Ханой минораси» масаласини кўрайлик.

**Масала.** Учта  $A$ ,  $B$ ,  $C$  қозик ва  $n$ -та ҳар хил ўлчамли халқалар мавжуд. Халқаларни ўлчамлари ўсиш тартибида 1 дан  $n$  гача тартибланган. Бошда барча халқалар  $A$  қозикқа 5.3а -расмдагидек жойлаштирилган.  $A$  қозикдаги барча халқаларни  $B$  қозикқа, ёрдамчи  $C$  қозикдан фойдаланган ҳолда, қуйидаги қоидаларга амал қилган ҳолда ўтказиш талаб этилади: халқаларни биттадан кўчириш керак ва катта ўлчамли халқани кичик ўлчамли халқа устига қўйиш мумкин эмас.



5.3-расм. Ханой минораси масаласини ечиш жараёни

Амаллар кетма-кетлигини чоп этадиган («Халқа  $q$  дан  $r$  га ўтказилсин») кўринишида, бунда  $q$  ва  $r$  - 5.3-расмдаги  $A, B$  ёки  $C$  халқалар. Берилган  $n$  та халқа учун масала ечилсин.

Кўрсатма: халқаларни  $A$  дан  $B$  га тўғри ўтказишда 5.3б –расмлар-даги ҳолат юзага келади, яъни  $n$  халқани  $A$  дан  $B$  ўтказиш масаласи  $n-1$  халқани  $A$  дан  $C$  га ўтказиш, ҳамда битта халқани  $A$  дан  $B$  га ўтказиш масаласига келади. Ундан кейин  $C$  қозикдаги  $n-1$  халқани  $A$  қозик ёрдамида  $B$  қозикқа ўтказиш масаласи юзага келади ва ҳақоза.

```
#include <iostream.h>
void Hanoy(int n, char a='A', char b='B', char
c='C')
{
    if (n)
    {
        Hanoy(n-1, a, c, b);
        cout<<"Xalqa"<<a<<" dan "<<b<<" ga
o' tkazilsin.\n";
        Hanoy(n-1, c, b, a);
    }
}
```

```

}
int main()
{
    unsigned int Xalqalar_Soni;
    cout<<"Hanoy minorasi masalasi"<<endl;
    cout<<"Xalqalar sonini kiriting: ";
    cin>>Xalqalar_Soni;
    Hanoy(Xalqalar_Soni);
    return 0;
}

```

Халқалар сони 3 бўлганда (*Xalqalar\_Soni=3*) программа экранга халқаларни кўчириш бўйича амаллар кетма-кетлигини чоп этади:

```

Xalqa A dan B ga o'tkazilsin.
Xalqa A dan C ga o'tkazilsin.
Xalqa B dan C ga o'tkazilsin.
Xalqa A dan B ga o'tkazilsin.
Xalqa C dan A ga o'tkazilsin.
Xalqa C dan B ga o'tkazilsin
Xalqa A dan B ga o'tkazilsin

```

Рекурсия чиройли, ихчам кўрингани билан хотирани тежаш ва ҳисоблаш вақтини қисқартириш нуқтаи-назаридан уни имкон қадар итератив ҳисоблаш билан алмаштирилгани маъқул. Масалан,  $x$  ҳақи-қий сонининг  $n$ -даражасини ҳисоблашнинг қуйидаги ечим варианти нисбатан кам ресурс талаб қилади ( $n$ - бутун ишорасиз сон):

```

double Butun_Daraja(double x,int n)
{
    double p=1;
    for(int i=1;i<=n;i++)p*=x;
    return p;
}

```

Иккинчи томондан, шундай масалалар борки, уларни ечишда рекурсия жуда самарали, ҳаттоки ягона усулдир. Хусусан, грамматик таҳлил масалаларида рекурсия жуда ҳам ўнғай ҳисобланди.

## 26-мавзу: Фойдаланувчи томонидан аниқланган берилганлар турлари

<b>Ажратилган соат:</b>	2 соат
<b>Машғулот тури:</b>	маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

1. C++ тилидаги программа тузилиши ва унинг компиляцияси
2. C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

### Санаб ўтилувчи тур

Кўп миқдордаги, мантиқан боғланган ўзгармаслардан фойдаланилганда санаб ўтилувчи турдан фойдаланилгани маъқул. Санаб ўтилувчи ўзгармаслар *enum* калит сўзи билан аниқланади. Мазмунли бўйича бу ўзгармаслар оддий бутун сонлардир. Санаб ўтилувчи ўзгармаслар C++ стандарти бўйича бутун турдаги ўзгармаслар ҳисоб-ланади. Ҳар бир ўзгармасга (сонга) мазмунли ном берилади ва бу идентификаторни программанинг бошқа жойларида номлаш учун ишлатилиши мумкин эмас. Санаб ўтилувчи тур қўйидаги кўринишга эга:

```
enum <санаб ўтиладиган тур номи> { <ном1> =<қиймат1>,  
    <ном2> =<қиймат2>, ... <номn> =<қийматn> };
```

Бу ерда, *enum* - калит сўз (инглизча *enumerate* - санамоқ); <санаб ўтиладиган тур номи>- ўзгармаслар рўйхатининг номи; <ном<sub>i</sub>> - бутун қийматли константаларнинг номлари; <қиймат<sub>i</sub>>- шарт бўлмаган инициализация қиймати (ифода).

Мисол учун ҳафта кунлари билан боғлиқ масала ечишда ҳафта кунларини *dush* (душанба), *sesht* (сешанба), *chor* (чоршанба), *paysh* (пайшанба), *juma* (жума), *shanba* (шанба), *yaksh* (якшанба) ўзгармас-ларини ишлатиш мумкин ва улар санаб ўтилувчи тур ёрдамида битта сатрда ёзилади:

```
enum
```

```
Hafta{dush, sesht, chor, paysh, juma, shanba, yaksh};
```

Санаб ўтилувчи ўзгармаслар қуйидаги хоссага эга: агар ўзгармас қиймати кўрсатилмаган бўлса, унинг қиймати олдинги ўзгармас қийматидан биттага ортиқ бўлади. Келишув бўйича биринчи ўзгармас қиймати 0 бўлади.

Инициализация ёрдамида ўзгармас қийматини ўзгартириш мумкин:

```
enum Hafta {dush=8, sesht, chor=12, paysh=13, juma=16,  
            shanba, yaksh=20};
```

Бу эълонда *sesht* қиймати 9, *shanba* эса 17 га тенг бўлади.

Санаб ўтилувчи ўзгармасларнинг номлари ҳар хил бўлиши керак, лекин уларнинг қийматлари бир хил бўлиши мумкин:

```
enum{ nol=0, toza=0, bir, ikki, juft=2, uch};
```

Ўзгармаснинг қиймати ифода кўринишда берилиши мумкин, фақат ифодадаги номларнинг қийматлари шу қадамдагача аниқланган бўлиши керак:

```
enum {ikki=2,turt=ikki*2};
```

Ўзгармаснинг қиймати манфий сон бўлиши ҳам мумкин:

```
enum {minus2=-2,minus1,nul,bir};
```

### Фойдаланувчи томонидан аниқланган берилганлар тури

C++ тилида фойдаланувчи томонидан нафақат структура ёки бирлашма турлари, балки айни пайтда мавжуд (аниқланган) турлар асосида янги турларни яратиши мумкин.

Фойдаланувчи томонидан аниқланадиган тур *typedef* калит сўзи билан бошланади, ундан кейин мавжуд тур кўрсатилади ва идентификатор ёзилади. Охирида ёзилган идентификатор - янги яратилган турнинг номи ҳисобланади. Масалан,

```
typedef unsigned char byte;
```

ифодаси *byte* деб номланувчи янги турни яратади ва ўз мазмунига кўра *unsigned char* тури билан эквивалент бўлади. Кейинчалик, программада хотирадан бир байт жой эгаллайдиган ва [0..255] оралиғидаги қийматларни қабул қиладиган *byte* туридаги ўзгарувчи (ўзгармасларни) эълон қилиш мумкин:

```
byte c=65;  
byte Byte=0xFF;
```

Массив кўринишидаги фойдаланувчи томонидан аниқланувчи тур эълони қуйидагича бўлади:

```
typedef char Ism[30];  
Ism ism;
```

*Ism* туридаги *ism* ўзгарувчиси эълони - бу 30 белгидан иборат массив (сатр) эълонидир.

Одатда ечилаётган масаланинг предмет соҳаси терминларида ишлаш учун структуралар қайта номланади. Натижада мураккаб тузилишга эга бўлган ва зарур хусусиятларни ўзига жамлаган янги турларни яратишга мувофиқ бўлинади.

Масалан, комплекс сон ҳақидаги маълумотларни ўз ичига оловчи *Complex* тури қуйидагича аниқланади:

```
typedef struct  
{  
    double re,im;  
} Complex;
```

Энди комплекс сон эълонини

```
Complex KSon;
```

ёзиш мумкин ва унинг майдонларига мурожаат қилиш мумкин:

```
KSon.re=5.64;  
KSon.im=2.3;
```

## 27-мавзу: Номлар фазоси

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

ДАРС Р Е Ж А С И (асосий саволлар):

1. C++ тилидаги программа тузилиши ва унинг компиляцияси
2. C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

### Номлар фазоси

Маълумки, программага кўшилган сарлавҳа файлларида эълон қилинган идентификатор ва ўзгармаслар компилятор томонидан ягона глобал номлар фазосига киритилади. Агар программа кўп миқдордаги сарлавҳа файлларни ишлатса ва ундаги идентификаторлар (функция номлари ва ўзгарувчилар номлари, синфлар номлари ва ҳакозалар), ўзгармаслар номлари турли программа тузувчилар томонидан мустақил равишда аниқланган бўлса, бир хил номларни ишлатиш билан боғлиқ муаммолар юзага келиш эҳтимоли катта бўлади. Номлар фазоси тушунчасини киритилиши мазкур муаммони маълум бир маънода ҳал қилишга ёрдам беради. Агар программада янги идентификаторни аниқлаши керак ва худди шу номни бошқа модулларда ёки кутубхоналарда ишлатиш хавфи бўладиган бўлса, бу идентификаторлар учун ўзининг шахсий номлар фазосини аниқлаш мумкин. Бунга *namespace* калит сўзидан фойдаланилган ҳолда эришилади:

```
namespace <номлар фазосининг номи>  
{  
    // эълонлар  
}
```

Номлар фазоси ичида эълон қилинган идентификаторлар фақат <номлар фазосининг номи> кўриниш соҳасида бўлади ва юзага келиши мумкин бўлган келишмовчиликларнинг олди олинади.

Мисол тариқасида қуйидаги номлар фазосини яратайлик:

```
namespace Shaxsiy_nomlar  
{
```

```
int x,y,z;
void Mening_funksiyam(char belgi);
}
```

Компиляторга конкрет номлар фазосидаги номларни ишлатиш кераклигини кўрсатиш учун кўриниш соҳасига рухсат бериш амали-дан фойдаланиш мумкин:

```
Shaxsiy_nomlar::x=5;
```

Агар программа матнида конкрет номлар фазосига нисбатан кўп мурожаат қилинадиган бўлса *using namespace* курилмасини ишлатиш орқали ёзувни соддалаштириш мумкин:

```
using namespace <номлар фазоси номи>;
```

Масалан,

```
using namespace Shaxsiy_nomlar;
```

кўрсатмаси компиляторга, бундан кейин токи навбатдаги *using* учрамагунча *Shaxsiy\_nomlar* фазосидаги номлар ишлатилиши керак-лигини билдиради:

```
x=0; y=z=10;
Mening_functsiyam('A');
```

Программа ва унга кўшилган сарлавҳа файллари томонидан аниқланадиган номлар фазоси *std* деб номланади. Стандарт фазога ўтиш керак бўлса

```
using namespace std;
```

кўрсатмаси берилади.

Агар бирорта номлар фазосидаги алоҳида бир номга мурожаат қилиш зарур бўлса, *using* курилмасини бошқа шаклида фойдаланилади. Мисол учун

```
using namespace std;
using namespace Shaxsiy_nomlar::x;
```

кўрсатмаси *x* идентификаторини *Shaxsiy\_nomlar* фазосидан ишлатиш кераклигини билдиради.

Шуни қайд этиш керакки, *using namespace* курилмаси стандарт номлар фазоси кўриниш соҳасини беркитади ва ундаги номга мурожаат қилиш учун кўриниш соҳасига рухсат бериш амалидан фойдаланиш зарур бўлади.

Номлар фазоси функция ичида эълон қилиниши мумкин эмас, лекин улар бошқа номлар фазоси ичида эълон қилиниши мумкин. Ичма-ич жойлашган номлар фазосидаги идентификаторга мурожаат қилиш учун уни қамраб олган барча номлар фазоси номлар кетма-кет равишда кўрсатилиши керак. Мисол учун, қуйидаги кўринишда номлар фазоси эълон қилинган бўлсин:

```

namespace Yuqori
{
    namespace Urta
    {
        namespace Ichki {int Ichki_n;}
    }
}

```

У ҳолда *Ichki\_n* ўзгарувчисига мурожаат қуйидаги кўринишда бўлади:

```

Yuqori::Urta::Ichki::Ichki_n=0;

```

Номлар фазосида функцияни эълон қилишда номлар фазосида фақат функция прототипини эълон қилиш ва функция танасини бошқа жойда эълон қилиш маъқул вариант ҳисобланади. Бу ҳолат-нинг кўринишига мисол:

```

namespace Nomlar_fazosi
{
    char c;
    int I;
    void Functsiya(char Bayroq);
}
...
void Nomlar_fazosi::Functsiya(char Bayroq)
{
    // функция танаси
}

```

Умуман олганда, ўз номига эга бўлмаган номлар фазосини эълон қилиш мумкин. Бу ҳолда *namespace* калит сўзидан кейин ҳеч нима ёзилмайди. Мисол учун

```

namespace
{
    char c_nomsiz;
    int i_nomsiz;
}

```

кўринишидаги номлар фазоси элементларига мурожаат ҳеч бир префикс ишлатмасдан амалга оширилади. Номсиз номлар фазоси фақат ўзи эълон қилинган файл чегарасида амал қилади.

C++ тили номлар фазосининг псевдонимларини аниқлаш имконини беради. Бу йўл орқали номлар фазосини бошқа ном билан ишла-тиш мумкин бўлади. Масалан, номлар фазоси номи узун бўлганда унга қисқа ном билан мурожаат қилиш:

```

namespace Juda_uzun_nomli_fazo {float y;}
Juda_uzun_nomli_fazo::y=0;

```



```
namespace Qisqa_nom=Juda_uzun_nomli_fazo;
Qisqa_nom::y=13.2;
```

### Жойлаштириладиган (inline) функциялар

Компилятор ишлаши натижасида ҳар бир функция машина коди кўринишида бўлади. Агар программада функцияни чақириш кўрсатмаси бўлса, шу жойда функцияни адреси бўйича чақиришнинг машина коди шаклланади. Одатда функцияни чақириш процессор томонидан кўшимча вақт ва хотира ресурсларини талаб қилади. Шу сабабли, агар чақириладиган функция ҳажми унчалик катта бўлмаган ҳолларда, компиляторга функцияни чақириш коди ўрнига функция танасини ўзини жойлаштиришга кўрсатма бериш мумкин. Бу иш функция прототипини *inline* калит сўзи билан эълон қилиш орқали амалга оширилади. Натижада ҳажми ошган, лекин нисбатан тез бажариладиган программа коди юзага келади.

Функция коди жойлаштириладиган программага мисол.

```
#include <iostream.h>
inline int Summa(int,int);
int main()
{
    int a=2,b=6,c=3;
    char yangi_qator='\n';
    cout<<Summa(a,b)<<yangi_qator;
    cout<<Summa(a,c)<<yangi_qator;
    cout<<Summa(b,c)<<yangi_qator;
    return 0;
}
int Summa(int x,int y){return x+y;}
```

Келтирилган программа кодини ҳосил қилишда *Summa()* функцияси чақирилган жойларга унинг танасидаги буйруқлар жойлаштирилади.

## 28-мавзу: Стандарт кутубхона функциялари

Ажратилган соат:	2 соат
Машғулот тури:	маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

- 1.C++ тилидаги программа тузилиши ва унинг компиляцияси
- 2.C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

## 29-мавзу: Кўрсаткичлар ва адрес олувчи ўзгарувчилар

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

1. C++ тилидаги программа тузилиши ва унинг компиляцияси
2. C++ тилида бажариловчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

### Кўрсаткичлар

Программа матнида ўзгарувчи эълон қилинганда, компилятор ўзгарувчига хотирадан жой ажратади. Бошқача айтганда, программа коди хотирага юкланганда берилганлар учун, улар жойлашадиган сегментнинг бошига нисбатан силжишини, яъни нисбий адресини аниқлайди ва объект код ҳосил қилишда ўзгарувчи учраган жойга унинг адресини жойлаштиради.

Умуман олганда, программадаги ўзгармаслар, ўзгарувчилар, функциялар ва синф объектлар адресларини хотиранинг алоҳида жойида сақлаш ва улар устидан амаллар бажариш мумкин. Қиймат-лари адрес бўлган ўзгарувчиларга *кўрсаткич ўзгарувчилар* дейилади.

Кўрсаткич уч хил турда бўлиши мумкин:

- бирорта объектга, хусусан ўзгарувчига кўрсаткич;
- функцияга кўрсаткич;
- *void* кўрсаткич.

Кўрсаткичнинг бу хусусиятлари унинг қабул қилиши мумкин бўлган қийматларида фарқланади.

Кўрсаткич албатта бирорта турга боғланган бўлиши керак, яъни у кўрсатган адресда қандайдир қиймат жойланиши мумкин ва бу қийматнинг хотирада қанча жой эгаллаши олдиндан маълум бўлиши шарт.

**Объектга кўрсаткич.** Бирор объектга кўрсаткич (шу жумладан ўзгарувчига). Бундай кўрсаткичда маълум турдаги (таянч ёки ҳосила-вий турдаги) берилганларнинг хотирадаги адреси жойлашади. Объектга кўрсаткич қуйидагича эълон қилинади:

<тур> \*<ном>;

Бу ерда <тур> - кўрсаткич аниқлайдиган адресдаги қийматнинг тури, <ном> - объект номи (идентификатор). Агар бир турда бир нечта

кўрсаткичлар эълон қилинадиган бўлса, ҳар бир кўрсаткич учун ‘\*’ белгиси қўйилиши шарт:

```
int *i,j,*k;  
float x,*y,*z;
```

Келтирилган мисолда *i* ва *k* - бутун турдаги кўрсаткичлар ва *j* - бутун турдаги ўзгарувчи, иккинчи операторда *x* - ҳақиқий ўзгарувчи ва *y,z* - ҳақиқий турдаги кўрсаткичлар эълон қилинган.

**void кўрсаткич.** Бу кўрсаткич объект тури олдиндан номаълум бўлганда ишлатилади. *void* кўрсаткичининг муҳим афзалликларидан бири - унга ҳар қандай турдаги кўрсаткич қийматини юклаш мумкин-лигидир. *void* кўрсаткич адресидаги қийматни ишлатишдан олдин, уни аниқ бир турга ошкор равишда келтириш керак бўлади. *void* кўрсаткични эълон қилиш қуйидагича бўлади:

```
void *<ном>;
```

Кўрсаткичнинг ўзи ўзгармас ёки ўзгарувчан бўлиши ва ўзгармас ёки ўзгарувчилар адресига кўрсатиши мумкин, масалан:

```
int i;                // бутун ўзгарувчи  
const int ci=1;      // бутун ўзгармас  
int * pi;             // бутун ўзгарувчига кўрсаткич  
const int *pci;       // бутун ўзгармасга кўрсаткич  
int *const cp=&i;     // бутун ўзгарувчига ўзгармас  
                    // кўрсаткич  
const int*const cps=&ci; // бутун ўзгармасга  
ўзгармас  
                    // кўрсаткич
```

Мисоллардан кўриниб турибдики, ‘\*’ ва кўрсаткич номи ора-сида турган *const* модификатори фақат кўрсаткичнинг ўзига тегишли ҳисобланади ва уни ўзгартириш мумкин эмаслигини билдиради, ‘\*’ белгисидан чапда турган *const* эса кўрсатилган адресдаги қиймат ўзгармас эканлигини билдиради.

Кўрсаткичга қийматни бериш учун ‘&’ - адресни олиш амали ишлатилади.

Кўрсаткич ўзгарувчиларининг амал қилиш соҳаси, яшаш даври ва кўриниш соҳаси умумий қоидаларга бўйсунди.

**Функцияга кўрсаткич.** Функцияга кўрсаткич программа жойлашган хотирадаги функция кодининг бошланғич адресини кўрса-тади, яъни функция чақирилганда бошқарув айна шу адресга узатилади. Кўрсаткич орқали функцияни оддий ёки воситали чақириш амалга ошириш мумкин. Бунда функция унинг номи бўйича эмас, балки функцияга кўрсатувчи ўзгарувчи орқали чақирилади. Функцияни бошқа функцияга аргумент сифатида узатиш ҳам функция кўрсаткичи орқали бажарилади. Функцияга кўрсаткичнинг ёзилиш синтаксиси қуйидагича:

<тур> (\* <ном>) (<параметрлар рўйхати>);

Бунда <тур>- функция қайтарувчи қиймат тури; \*<ном> - кўрсаткич ўзгарувчининг номи; <параметрлар рўйхати> - функция параметрларининг ёки уларнинг турларининг рўйхати.

Масалан:

```
int (*fun)(float,float);
```

Бу ерда бутун сон турида қиймат қайтарадиган *fun* номидаги функцияга кўрсаткич эълон қилинган ва у иккита ҳақиқий турдаги параметрларга эга.

**Масала.** Берилган бутун  $n=100$  ва  $a, b$  - ҳақиқий сонлар учун  $f_1(x) = 5 \sin(3x) + x$ ,  $f_2(x) = \cos(x)$  ва  $f_3(x) = x^2 + 1$  функциялар учун

$\int_a^b f(x) dx$  интегралини тўғри тўртбурчаклар формуласи билан тақрибан

ҳисоблансин:

$$\int_a^b f(x) dx \approx h [f(x_1) + f(x_2) + \dots + f(x_n)],$$

бу ерда  $h = \frac{b-a}{n}$ ,  $x_i = a + ih - h/2, i = 1..n$ .

Программа бош функция, интеграл ҳисоблаш ва иккита математик функциялар -  $f_1(x)$  ва  $f_3(x)$  учун аниқланган функциялардан ташкил топади,  $f_2(x) = \cos(x)$  функциянинг адреси «math.h» сарлавҳа файли-дан олинади. Интеграл ҳисоблаш функциясига кўрсаткич орқали интеграл ҳисобланадиган функция адреси,  $a$  ва  $b$  - интеграл чегара-лари қийматлари узатилади. Ораликни бўлишлар сони -  $n$  глобал ўзгармас қилиб эълон қилинади.

```
#include <iostream.h>
#include <math.h>
const int n=100;
double f1(double x){return 5*sin(3*x)+x;}
double f3(double x){return x*x+1;}
double      Integral(double(*f)(double),double
a,double b)
{
    double x,s=0;
    double h=(b-a)/n;
    x=a-h/2;
    for(int i=1;i<=n;i++) s+=f(x+=h);
    s*=h;
    return s;
}
int main()
```

```

{
    double a,b;
    int menu;
    while(1)
    {
        cout<<"\n Integral osti funksiyasiyalari:\n";
        cout<<"\t1:f1(x)=5*sin(3*x)+x\n";
        cout<<"\t2:f2(x)=cos(x)\n";
        cout<<"\t3:f3(x)=x^2+1\n";
        do
        {
            cout<<"Funksiya nomerini tanlang (0-chiqish)-
>";
            cin>>menu;
        }
        while (menu<0||menu>3);
        if(!menu)break;
        cout<<"Integral oralig'ining quyi chegarasi
a=";
        cin>>a;
        cout<<"Integral oralig'ining yuqori chegarasi
b=";
        cin>>b;
        cout<<"Funksiya integrali S=";
        switch (menu)
        {
            case 1:cout<<Integral(f1,a,b)<<endl;break;
            case 2:cout<<Integral(cos,a,b)<<endl;break;
            case 3:cout<<Integral(f3,a,b)<<endl;
        }
    }
    return 0;
}

```

Программанинг иши чексиз такрорлаш оператори танасини бажаришдан иборат. Такрорлаш танасида фойдаланувчига иш режими-мини танлаш бўйича меню таклиф қилинади:

**Integral osti funksiyasiyalari:**

1:f1(x)=5\*sin(3\*x)+x

2:f2(x)=cos(x)

3:f3(x)=x^2+1

**Funksiya nomerini tanlang (0-chiqish)->**

Фойдаланувчи 0 ва 3 оралиғидаги бутун сонни киритиши керак. Агар киритилган сон (*menu* ўзгарувчи қиймати) 0 бўлса, *break* оператори ёрдамида такрорлашдан, кейин программадан чиқилади. Агар *menu*

қиймати 1 ва 3 оралғида бўлса, интегралнинг қуйи ва юқори чегараларини киритиш сўралади, ҳамда *Integral()* функцияси мос функция адреси билан чақирилади ва натижа чоп этилади. Шунга эътибор бериш керакки, интеграл чегараларининг қийматларини тўғри киритилишига фойдаланувчи жавобгар.

### Кўрсаткичга бошланғич қиймат бериш

Кўрсаткичлар кўпинча динамик хотира (бошқача номи «уюм» ёки «*heap*») билан боғлиқ ҳолда ишлатилади. Хотиранинг динамик дейилишига сабаб, бу соҳадаги бўш хотира программа ишлаш жараёнида, керакли пайтида ажратиб олинади ва зарурат қолмаганида қайтарилади (бўшатилади). Кейинчалик, бу хотира бўлаги программа томонидан бошқа мақсадда яна ишлатилиши мумкин. Динамик хотирага фақат кўрсаткичлар ёрдамида мурожаат қилиш мумкин. Бундай ўзгарувчилар *динамик ўзгарувчилар* дейилади ва уларни яшаш вақти яратилган нуқтадан бошлаб программа охиригача ёки ошкор равишда йўқотилган (боғланган хотира бўшатирилган) жойгача бўлади.

Кўрсаткичларни эълон қилишда унга бошланғич қийматлар бериш мумкин. Бошланғич қиймат (инициализатор) кўрсаткич номи-дан сўнг ёки қавс ичида ёки '=' белгидан кейин берилади. Бошланғич қийматлар қуйидаги усуллар билан берилиши мумкин:

1. Кўрсаткичга мавжуд бўлган объектнинг адресини бериш:

а) адресни олиш амал орқали:

```
int i=5,k=4; // бутун ўзгарувчилар
int *p=&i;   // p кўрсаткичга i ўзгарувчининг
              // адреси ёзилади
int *p1(&k); // p1 кўрсаткичга k ўзгарувчининг
              // адреси ёзилади
```

б) бошқа, инициализацияланган кўрсаткич қийматини бериш:

```
int * r=p; // p олдин эълон қилинган ва қийматга
эга
          // бўлган кўрсаткич
```

в) массив ёки функция номини бериш:

```
int b[10];      // массивни эълон қилиш
int *t=b;       // массивнинг бошланғич адресини
бериш
void f(int a){/* ... */} // функцияни аниқлаш
void (*pf)(int); // функцияга кўрсаткични эълон
қилиш
pf=f;           // функция адресини кўрсаткичга бериш
```

2. Ошкор равишда хотиранинг абсолют адресини бериш:

```
char *vp =(char *)0xB8000000;
```

Бунда 0xB8000000 - ўн олтилик ўзгармас сон ва (*char\**) - турга келтириш амали бўлиб, у *vp* ўзгарувчисини хотиранинг абсолют адресидаги байтларни *char* сифатида қайта ишловчи кўрсаткич турига айлантирилишини англатади.

3. Бўш қиймат бериш:

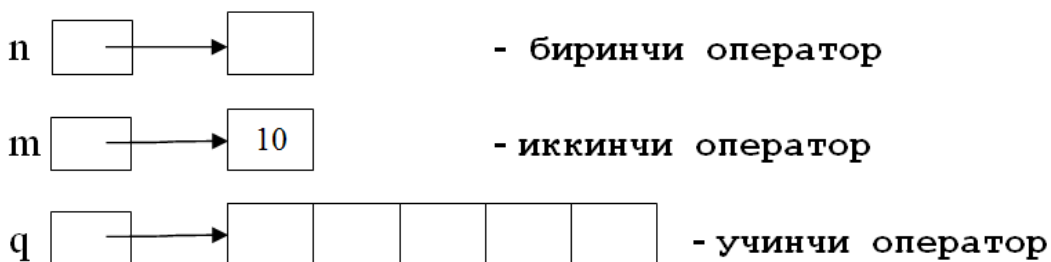
```
int *vector=NULL;
int *r=0;
```

Биринчи сатрда махсус *NULL* ўзгармаси ишлатилган, иккинчи сатрда 0 қиймат ишлатилган. Иккала ҳолда ҳам кўрсаткич ҳеч қандай объектга мурожаат қилмайди. Бўш кўрсаткич асосан кўрсаткични аниқ бир объектга кўрсатаётган ёки йўқлигини аниқлаш учун ишла-тилади.

4. Динамик хотирада *new* амали билан жой ажратиш ва уни адресини кўрсаткичга бериш:

```
int * n=new int;           // биринчи оператор
int * m=new int(10);       // иккинчи оператор
int * q=new int[5];        // учинчи оператор
```

Биринчи операторда *new* амали ёрдамида динамик хотирада *int* учун етарли жой ажратиб олиниб, унинг адреси *n* кўрсаткичга юкланади. Кўрсаткичнинг ўзи учун жой компиляция вақтида ажра-тилади.



6.1-расм. Динамик хотирадан жой ажратиш

Иккинчи операторда жой ажратишдан ташқари *m* адресига бошланғич қиймат - 10 сонини жойлаштиради.

Учинчи операторда *int* туридаги 5 элемент учун жой ажра-тилган ва унинг бошланғич адреси *q* кўрсаткичга берилаяпти.

Хотира *new* амали билан ажратилган бўлса, у *delete* амали билан бўшатилиши керак. Юқоридаги динамик ўзгарувчилар билан боғлан-ган хотира қуйидагича бўшатилади:

```
delete n; delete m; delete[]q;
```

Агарда хотира *new[]* амали билан ажратилган бўлса, уни бўшатиш учун *delete[]* амалини ўлчови кўрсатилмаган ҳолда қўллаш керак.

Хотира бўшатирилганлигига қарамадан кўрсаткични ўзини кейинчалик қайта ишлатиш мумкин.

## Кўрсаткич устида амаллар

Кўрсаткич устида қуйидаги амаллар бажарилиши мумкин:

- 1) объектга воситали мурожаат қилиш амали;
- 2) қиймат бериш амали;
- 3) кўрсаткичга ўзгармас қийматни қўшиш амали;
- 4) айириш амали;
- 5) инкремент ва декремент амаллари;
- 6) солиштириш амали;
- 7) турга келтириш амали.

Воситали мурожаат қилиш амали кўрсаткичдаги адрес бўйича жойлашган қийматни олиш ёки қиймат бериш учун ишлатилади:

```
char a;          // char туридаги ўзгарувчи
эълони.
char *p=new char; // Кўрсаткични эълон қилиб, унга
                  // динамик хотирадан ажратилган
                  // хотиранинг адресини бериш
*p='b'; // p адресига қиймат жойлаштириш
a=*p;   // a ўзгарувчисига p адресидаги қийматни
бериш
```

Шуни қайд қилиб ўтиш керакки, хотиранинг аниқ бир жойидаги адресни бир пайтнинг ўзида бир нечта ва ҳар хил турдаги кўрсаткич-ларга бериш мумкин ва улар орқали мурожаат қилинганда берилган-нинг ҳар хил турдаги қийматларини олиш мумкин:

```
unsigned long int A=0Xcc77ffaa;
unsigned short int * pint=(unsigned short
int*)&A;
unsigned char * pchar=(unsigned char*)&A;
cout<<hex<<A<<' | '<<hex<<*pint<<' | '<<hex<<(int)*pchar;
char;
```

Экранга ҳар хил қийматлар чоп этилади:

```
cc77ffaa|ffaa|aa
```

Ўзгарувчилар битта адресда жойлашган ҳолда яхлит қийматнинг турли бўлақларини ўзлаштиради. Бунда, бир байтдан катта жой эгал-лаган сон қийматининг хотирада «*тескари*» жойлашиши инобатга олиниши керак.

Агар ҳар хил турдаги кўрсаткичларга қийматлар берилса, албат-та турга келтириш амалидан фойдаланиш керак:

```
int n=5;
float x=1.0;
int *pi=&n;
float *px=&x;
```



```

void *p;
int *r,*r1;
px=(float*)&n;
p=px;
r=(int*)p;
r1=pi;

```

Кўрсаткич турини *void* турига келтириш амалда маънога эга эмас. Худди шундай, турлари бир хил бўлган кўрсаткичлар учун турни келтириш амалини бажаришга ҳожат йўқ.

Кўрсаткич устидан бажариладиган арифметик амалларда авто-матик равишда турларнинг ўлчами ҳисобга олинади.

Арифметик амаллар фақат бир хил турдаги кўрсаткичлар устидан бажарилади ва улар асосан, массив тузилмаларига кўрсаткич-лар устида бажарилади.

Инкремент амали кўрсаткични массивнинг кейинги элементиға, декремент эса аксинча, битта олдинги элементининг адресига кўчиради. Бунда кўрсаткичнинг қиймати *sizeof*(<массив элементи-нинг тури>) қийматига ўзгаради. Агар кўрсаткич *k* ўзгармас қийматга оширилса ёки камайтирилса, унинг қиймати *k\*sizeof*(<массив элементининг тури>) катталиққа ўзгаради.

Масалан:

```

short int * p=new short[5];
long * q=new long[5];
p++;      // p қиймати 2 ошади
q++;      // q қиймати 4 га ошади
q+=3;     // q қиймати 3*4=12 ошади

```

Кўрсаткичларнинг айирмаси деб, улар айирмасининг тур ўлча-мига бўлинишига айтилади. Кўрсаткичларни ўзаро қўшиш мумкин эмас.

### Адресни олиш амали

Адресни олиш қуйидагича эълон қилинади:

```
<тур> & <ном>;
```

Бу ерда <тур> - адреси олинadиган қийматнинг тури, <ном>- адрес олувчи ўзгарувчи номи. Ўртадаги ‘&’ белгисига *адресни олиш амали* дейилади.

Бу кўринишда эълон қилинган ўзгарувчи шу турдаги ўзгарувчи-нинг синоними деб қаралади. Адресни олиш амали орқали битта ўзгарувчига ҳар хил ном билан мурожаат қилиш мумкин бўлади.

Мисол:

```

int kol;
int & pal=kol;    // pal ўзгарувчиси kol
                  // ўзгарувчисининг альтернатив
                  номи

```

```
const char & cr='\n' ;//cr ўзгармас литерли  
ўзгармас  
//адресини олади
```

Адресни олиш амалини ишлатишда қуйидаги қоидаларга риоя қилиш керак: адрес олувчи ўзгарувчи функция параметри сифатида ишлатилган ёки *extern* билан тавсифланган ёки синф майдонига мурожаат қилингандан ҳолатлардан ташқари барча ҳолатларда бошланғич қийматга эга бўлиши керак.

Адресни олиш амали асосан функцияларда адрес орқали узатиловчи параметрлар сифатида ишлатилади.

Адрес олувчи ўзгарувчининг кўрсаткичдан фарқи шундаки, у алоҳида хотирани эгалламайди ва фақат ўз қиймати бўлган ўзгарувчи-нинг бошқа бир номи сифатида ишлатилади.

### 30-мавзу: Динамик массивлар

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

- 1.C++ тилидаги программа тузилиши ва унинг компиляцияси
- 2.C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

#### Динамик массивлар билан ишлаш

Статик массивларнинг камчиликлари шундаки, уларнинг ўлчамлари олдиндан маълум бўлиши керак, бундан ташқари бу ўлчамлар берилганларга ажратилган хотира сегментининг ўлчами билан чегараланган. Иккинчи томондан, етарлича катта ўлчамдаги массив эълон қилиб, конкрет масала ечилишида ажратилган хотира тўлиқ ишлатилмаслиги мумкин. Бу камчиликлар динамик массивлар-дан фойдаланиш орқали бартараф этилади, чунки улар программа ишлаши жараёнида керак бўлган ўлчамдаги массивларни яратиш ва зарурат қолмаганда йўқотиш имкониятини беради.

Динамик массивларга хотира ажратиш учун *malloc()*, *calloc()* функцияларидан ёки *new* операторидан фойдаланиш мумкин. Динамик объектга ажратилган хотирани бўшатиш учун *free()* функцияси ёки *delete* оператори ишлатилади.

Кўп ўлчамли динамик массивлар хотирада узлуксиз соҳани эгаллаши ёки тарқоқ ҳолда жойланиши мумкин. Одатда *malloc()*, *calloc()* функцияларидан узлуксиз соҳада жойлашган массивларни яратишда фойдаланилади, *new* операторидан тарқоқ ҳолда жойланиши мумкин бўлган массивлар яратилади.

Юқорида қайд қилинган функциялар “*alloc.h*” кутубхонасида жойлашган.

*malloc()* функциясининг синтаксиси

```
void * malloc(size_t size) ;
```

кўринишида бўлиб, у хотиранинг уюм қисмидан *size* байт ўлчамидаги узлуксиз соҳани ажратади. Агар хотира ажратиш муваффақиятли бўлса, *malloc()* функцияси ажратилган соҳанинг бошланиш адресини қайтаради. Талаб қилинган хотирани ажратиш муваффақиятсиз бўлса, функция *NULL* қийматини қайтаради.

Синтаксисдан кўриниб турибдики, функция *void* туридаги қиймат қайтаради. Амалда эса конкрет турдаги объект учун хотира ажратиш зарур бўлади. Бунинг учун *void* турини конкрет турга келтириш технологиясидан фойдаланилади. Масалан, бутун турдаги узунлиги 3 га тенг массивга жой ажратишни қуйидагича амалга ошириш мумкин:

```
int * pInt=(int*)malloc(3*sizeof(int)) ;
```

*calloc()* функцияси *malloc()* функциясидан фарқли равишда массив учун жой ажратишдан ташқари массив элементларини 0 қиймати билан инициализация қилади. Бу функция синтаксиси

```
void * calloc(size_t num, size_t size) ;
```

кўринишда бўлиб, *num* параметри ажратилган соҳада нечта элемент борлигини, *size* ҳар бир элемент ўлчамини билдиради.

*free()* хотирани бўшатиш функцияси ўчириладиган хотира бўла-гига кўрсаткич бўлган ягона параметрга эга бўлади:

```
void free(void * block) ;
```

*free()* функцияси параметрининг *void* турида бўлиши ихтиёрий турдаги хотира бўлагини ўчириш имконини беради.

Қуйидаги программада 10 та бутун сондан иборат динамик массив яратиш, унга қиймат бериш ва ўчириш амаллари бажарилган.

```
#include <iostream.h>  
#include <alloc.h>  
int main()  
{  
    int * pVector;  
    if ( (pVector=(int*)malloc(10*sizeof(int))) ==NULL)  
    {  
        cout<<"Xotira etarli emas!!!";  
    }  
}
```

```

    return -1;
}
// ажратилган хотира соҳасини тўлдириш
for(int i=0;i<10;i++) *(pVector+i)=i;
// вектор элементларини чоп этиш
for(int i=0;i<10;i++) cout<<*(pVector+i)<<endl;
// ажратилган хотира бўлагини қайтариш (ўчириш)
free(pVector);
return 0;
}

```

Кейинги программада  $n \times n$  ўлчамли ҳақиқий сонлар массиви-нинг бош диагоналидан юқорида жойлашган элементлар йиғинди-сини ҳисоблаш масаласи ечилган.

```

#include <iostream.h>
#include <alloc.h>
int main()
{
    int n;
    float * pMatr,s=0;
    cout<<"A(n,n): n=";
    cin>>n;

    if((pMatr=(float*)malloc(n*n*sizeof(float)))==NULL)
    {
        cout<<"Xotira etarli emas!!!";
        return -1;
    }
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)cin>>*(pMatr+i*n+j);
    for(int i=0;i<n;i++)
        for(int j=i+1;j<n;j++)s+=*(pMatr+i*n+j);
    cout<<"Matritsa bosh diagonalidan yuqoridagi ";
    cout<<"elementlar yig`indisi S="<<s<<endl;
    return 0;
}

```

*new* оператори ёрдамида массивга хотира ажратишда объект туридан кейин квадрат кавс ичида объектлар сони кўрсатилади. Масалан, бутун турдаги 10 та сондан иборат массивга жой ажратиш учун

```
pVector=new int[10];
```

ифодаси ёзилиши керак. Бунга қарама-қарши равишда, бу усулда ажратилган хотирани бўшатиш учун

```
delete [] pVector;
```

кўрсатмасини бериш керак бўлади.

Икки ўлчамли динамик массивни ташкил қилиш учун

```
int **a;
```

кўринишидаги «*кўрсаткичга кўрсаткич*» ишлатилади.

Бошда массив сатрлари сонига қараб кўрсаткичлар массивига динамик хотирадан жой ажратиш керак:

```
a=new int*[m] // бу ерда m массив сатрлари сони
```

Кейин, ҳар бир сатр учун такрорлаш оператори ёрдамида хотира ажратиш ва уларнинг бошланғич адресларини *a* массив элемент-ларига жойлаштириш зарур бўлади:

```
for(int i=0;i<m;i++)a[i]=new int[n]; //n устунлар сони
```

Шуни қайд этиш керакки, динамик массивнинг ҳар бир сатри хотиранинг турли жойларида жойлашиши мумкин (7.1 ва 7.3-расмлар).

Икки ўлчамли массивни ўчиришда олдин массивнинг ҳар бир элементи (сатри), сўнгга массивнинг ўзи йўқотилади:

```
for(i=0;i<m;i++) delete[]a[i];  
delete[]a;
```

Матрицани векторга кўпайтириш масаласи учун динамик массивлардан фойдаланишга мисол:

```
void main ()  
{  
    int n,m;  
    int i,j; float s;  
    cout<<"\nn="; cin>>n; // матрица сатрлари сони  
    cout<<"\nm="; cin>>m; // матрица устунлари сони  
    float *b=new float[m];  
    float *c=new float[n];  
    // кўрсаткичлар массивига хотира ажратиш  
    float **a=new float*[n] ;  
    for(i=0;i<n;i++) // ҳар бир сатр учун  
        a[i]=new float[m]; //динамик хотира ажратиш  
    for(j=0;j<m;j++)cin>>b[j];  
    for(i=0;i<n;i++)  
        for(j=0;j<m;j++)cin>>a[i][j];  
    for(i=0;i<n;i++)  
    {  
        for(j=0,s=0;j<m;j++)s+=a[i][j]*b[j];  
        c[i]=s;  
    }  
    for(i=0;i<n;i++)cout<<"\t c["<<i<<"]="<<c[i];
```

```

delete[]b;
delete[]c;
for (i=0;i<n;i++)
    delete[]a[i];
delete[]a;
return;
}

```

### 31-мавзу: Функция ва массивлар

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

ДАРС Р Е Ж А С И (асосий саволлар):

1. C++ тилидаги программа тузилиши ва унинг компиляцияси
2. C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

#### Функция ва массивлар

Функциялар массивни параметр сифатида ишлатиши ва уни функциянинг натижаси сифатида қайтариши мумкин.

Агар массив параметр орқали функцияга узатилса, элементлар сонини аниқлаш муаммоси туғилади, чунки массив номидан унинг узунлигини аниқлашнинг иложи йўқ. Айрим ҳолларда, масалан, белгилар массиви сифатида аниқланган сатр (ASCIIZ сатрлар) билан ишлаганда массив узунлигини аниқлаш мумкин, чунки сатрлар '\0' белгиси билан тугайди.

Мисол учун:

```

#include <iostream.h>
int  satr_uzunligi(char  s[])//массив  параметр
сифатида
{
    int m=0;
    while(s[m++] );
    return m-1;
}
void main()
{
    char z[]="Ushbu satr uzunligi = ";
    cout<<z<<satr_uzunligi(z) ;
}

```

```
}
```

Функция параметри сатр бўлмаган ҳолларда фиксирланган узунликдаги массивлар ишлатилади. Агар турли узунликдаги массивларни узатиш зарур бўлса, массив ўлчамларини параметр сифатида узатиш мумкин ёки бу мақсадда глобал ўзгарувчидан фойдаланишга тўғри келади.

Мисол:

```
#include <iostream.h>
float yigindi(int n,float *x) //бу иккинчи усул
{
    float s=0;
    for(int i=0;i<n;i++)s+=x[i];
    return s;
}
void main()
{
    float E[]={1.2,2.0,3.0,4.5,-4.0};
    cout<<yigindi(5,E);
}
```

Массив номи кўрсаткич бўлганлиги сабабли массив элементларининг қийматлари функция ичида ўзгартирилса, бу ўзгаришлар функциядан чиққандан кейин ҳам сақланиб қолади.

```
#include <iostream.h>
void vector_01(int n,int*x,int*y) //бу иккинчи
усул
{
    for(int i=0;i<n;i++)
        y[i]=x[i]>0?1:0;
}
void main()
{
    int a[]={1,2,-4,3,-5,0,4};
    int c[7];
    vector_01(7,a,c);
    for(int i=0;i<7;i++) cout<<'\\t'<<c[i]; }

```

**Масала.** Бутун турдаги ва элементлари камаймайдиган ҳолда тартибланган бир ўлчамли иккита массивларни ягона массивга, тартиб сақланган ҳолда бирлаштирилсин.

Программа матни:

```
#include <iostream.h>
\\бутун турдаги массивга кўрсаткич қайтарадиган
\\функция
int * massiv_ulas(int,int*,int,int*);
void main()
```

```

{
    int c[]={-1,2,5,9,10},d[]={1,7,8};
    int * h;
    h=massiv_ulash(5,c,3,d);
    for(int i=0;i<8;i++)cout<<' \t'<<h[i];
    delete[]h;
}
int * massiv_ulash(int n,int *a ,int m,int *b);
{
    int * x=new int[n+m];
    int ia=0,ib=0,ix=0;
    while (ia<n && ib<m)
        a[ia]>b[ib]?x[ix++]=b[ib++]:x[ix++]=a[ia++];
    while(ib<m)x[ix++]=b[ib++];
    while(ia<n)x[ix++]=a[ia++];
    return x;
}

```

Программа ишлаши натижасида экранга

```

-1  1    2    5    7    8    9    10

```

сонлар кетма-кетлиги чоп этилади.

Кўп ўлчамли массивлар билан ишлаш маълум бир мураккабликка эга, чунки массивлар хотирада жойлашуви турли вариантда бўлиши мумкин. Масалан, функция параметрлар рўйхатида  $n \times n$  ўлчамдаги ҳақиқий турдаги  $x[n][n]$  массивга мос келувчи параметрни

```
float sum(float x[n][n]);
```

кўринишда ёзиб бўлмайди. Муаммо ечими - бу массив ўлчамини параметр сифатида узатиш ва функция сарлавҳасини қуйидагича ёзиш керак:

```
float sum(int n,float x[][]);
```

Кўп ўлчамли массивларни параметр сифатида ишлатишда бир нечта усуллардан фойдаланиш мумкин.

*1-усул.* Массивнинг иккинчи ўлчамини ўзгармас ифода (сон) билан кўрсатиш:

```

float sum(int n,float x[][10])
{
    float s=0.0;
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
            s+=x[i][j];
    return s;
}

```



2-усул. Икки ўлчамли массив кўрсаткичлар массиви кўри-нишида аниқланган ҳолатлар учун кўрсаткичлар массивини (матрица сатрлар адресларини) бериш орқали:

```
float Summa(int n,float*p[])
{
    float s=0.0;
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
            s+=p[i][j];
    return s;
}
void main()
{
    float x[][4]={ {11,-12,13,14},{21,22,23,24},
                   {31,32,33,34},{41,42,43,44}};
    float *matr[4];
    for(int i=0;i<4;i++)matr[i]=(float *)&x[i];
    cout<<Summa(4,matr)<<endl;
}
```

3-усул. Кўрсаткичларга кўрсаткич кўринишида аниқланган динамик массивларни ишлатиш билан:

```
float Summa(int n,float**x)
{
    float s=0.0;
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)s+=x[i][j];
    return s;
}
void main()
{
    float **matr;
    int n;
    cin>>n;
    matr=new float *[n];
    for(int i=0;i<n;i++)
    {
        matr[i]=new float[n];
        for(int j=0;j<n;j++)
            matr[i][j]=(float)((i+1)*10+j);
    }
    cout<<Summa(n,matr);
    for(int i=0;i<n;i++) delete[]matr[i];
    delete[]matr;
}
```

Навбатдаги программада функция томонидан натижа сифатида икки ўлчамли массивни қайтаришига мисол келтирилган. Массив элементларнинг қийматлари тасодифий сонлардан ташкил топади. Тасодифий сонлар «math.h» кутубхонасидаги *random()* функция ёрдамида ҳосил қилинади:

```
#include <iostream.h>
#include <math.h>
int **rand_matr(int n,int m)
{
    int ** matr;
    matr=new int*[n];
    for(int i=0;i<n;i++)
    {
        matr[i]=new int[m];
        for(int j=0;j<m;j++) matr[i][j]=random(100);
    }
    return matr;
}
int Summa(int n,int m,int**ix)
{
    float s=0;
    for(int i=0;i<n;i++)
        for(int j=0;j<m;j++) s+=ix[i][j];
    return s;
}
void main()
{
    int n,m;
    cin>>n>>m;
    int **matrisa;
    randomize();
    matrisa=rand_matr(n,m);
    for(int i=0;i<n;i++)
    {
        cout<<endl<<i<<" - satr:"
        for (int j=0;j<m;j++) cout<<'\\t'<<matrisa[i][j];
    }
    cout<<endl<<"Summa="<<sum(n,m,matrisa);
    for(int i=0;i<n;i++) delete[]matrisa[i];
    delete[]matrisa;
}
```

**32-мавзу:** Сатрлар. Сатр устида амаллар.

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

### ДАРС Р Е Ж АСИ (асосий саволлар):

1. C++ тилидаги программа тузилиши ва унинг компиляцияси
2. C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

### Белги ва сатрлар

Стандарт C++ тили икки хилдаги белгилар мажмуасини қўллаб-қувватлайди. Биринчи тоифага, анъанавий, «*тор*» белгилар деб номланувчи 8-битли белгилар мажмуаси киради, иккинчисига 16-битли «*кенг*» белгилар киради. Тил кутубхонасида ҳар бир гуруҳ белгилари учун махсус функциялар тўплами аниқланган.

C++ тилида сатр учун махсус тур аниқланмаган. Сатр *char* тури-даги белгилар массиви сифатида қаралади ва бу белгилар кетма-кетлиги *сатр терминатори* деб номланувчи 0 кодли белги билан тугайди ('\0'). Одатда, нол-терминатор билан тугайдиган сатрларни *ASCIIZ-сатрлар* дейилади.

Қуйидаги жадвалда C++ тилида белги сифатида ишлатилиши мумкин бўлган ўзгармаслар тўплами келтирилган.

8.1-жадвал. C++ тилидаги белги ўзгармаслар

Белгилар синфлари	Белги ўзгармаслар
Катта ҳарфлар	'A' ... 'Z', 'А' ... 'Я'
Кичик ҳарфлар	'a' ... 'z', 'а' ... 'я'
Рақамлар	'0' ... '9'
Бўш жой	горизонтал табуляция (ASCII коди 9), сатрни ўтказиш (ASCII коди 10), вертикал табуляция (ASCII коди 11), формани ўтказиш (ASCII коди 12), кареткани қайтариш (ASCII коди 13)
Пунктуация белгилари (ажратувчилар)	! " # \$ % & ' ( ) * + , . / : ; < = > ? @ [ \ ] ^ _ {   } ~
Бошқарув белгилари	ASCII коди 0...1Fh оралиғида ва 7Fh бўлган белгилар
Пробел	ASCII коди 32 бўлган белги
Ўн олтилик рақамлар	'0' ... '9', 'A' ... 'F', 'a' ... 'f'

Сатр массиви эълон қилинишида, сатр охирига терминатор қўйилиши ва натижада сатрга қўшимча битта байт бўлишини инобатга олиниши керак:

```
char satr[10];
```

Ушбу эълонда *satr* сатри учун жами 10 байт ажратилади - 9 сатр ҳосил қилувчи белгилар учун ва 1 байт терминатор учун.

Сатр ўзгарувчилар эълон қилинишида бошланғич қийматларни қабул қилиши мумкин. Бу ҳолда компилятор автоматик равишда сатр узунлиги ҳисоблайди ва сатр охирига терминаторни қўшиб қўяди:

```
char Hafta_kuni []="Juma";
```

Ушбу эълон қуйидаги эълон билан эквивалент:

```
char Hafta_kuni []={'J','u','m','a','\0'};
```

Сатр қийматини ўқишда оқимли ўқиш оператори ">>" ўрнига *getline()* функциясини ишлатган маъқул ҳисобланади, чунки оқимли ўқишда пробеллар инкор қилинади (гарчи улар сатр белгиси ҳисоб-ланса ҳам) ва ўқиладиган белгилар кетма-кетлиги сатрдан «*ошиб*» кетганда ҳам белгиларни киритиш давом этиши мумкин. Натижада сатр ўзига ажратилган ўлчамдан ортиқ белгиларни «*қабул*» қилади. Шу сабабли, *getline()* функцияси иккита параметрга эга бўлиб, биринчи параметр ўқиш амалга ошириладиган сатрга кўрсаткич, иккинчи параметрда эса ўқиши керак бўлган белгилар сони кўрсатилади. Сатрни *getline()* функцияси орқали ўқишга мисол кўрайлик:

```
#include <iostream.h>  
int main()  
{  
    char satr[6];  
    cout<<"Satrni kiriting: "<<"\n";  
    cin.getline(satr,6);  
    cout<<"Siz kiritgan satr: "<<satr;  
    return 0;  
}
```

Программада ишлатилган *satr* сатри 5 та белгини қабул қилиши мумкин, ортиқчалари ташлаб юборилади. *getline()* функциясига мурожаатда иккинчи параметр қиймати ўқиладиган сатр узунлигидан катта бўлмаслиги керак.

Сатр билан ишлайдиган функцияларнинг аксарияти «*string.h*» кутубхонасида жамланган. Нисбатан кўп ишлатиладиган функциялар-нинг тавсифини келтирамиз.

### 33-мавзу: Сатр функциялари.

Ажратилган соат:	2 соат
Машғулот тури:	маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

1. C++ тилидаги программа тузилиши ва унинг компиляцияси
2. C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

### Сатр узунлигини аниқлаш функциялари

Сатрлар билан ишлашда, аксарият ҳолларда сатр узунлигини билиш зарур бўлади. Бунинг учун «string.h» кутубхонасида *strlen()* функцияси аниқланган бўлиб, унинг синтаксиси қуйидагича бўлади:

**size\_t strlen(const char\*string)**

Бу функция узунлиги ҳисобланиши керак бўлган сатр бошига кўрсаткич бўлган ягона параметрга эга ва у натижа сифатида ишорасиз бутун сонни қайтаради. *strlen()* функцияси сатрнинг реал узун-лигидан битта кам қиймат қайтаради, яъни нол-терминатор ўрни ҳисобга олинмайди.

Худди шу мақсадда *sizeof()* функциясидан ҳам фойдаланиш мумкин ва у *strlen()* функциясидан фарқли равишда сатрнинг реал узунлигини қайтаради. Қуйида келтирилган мисолда сатр узунлигини ҳисоблашнинг ҳар иккита варианты келтирилган:

```
#include <iostream.h>
#include <string.h>
int main()
{
    char Str[]="1234567890";
    cout<<"strlen(Str)="<<strlen(Str)<<endl;
    cout<<"sizeof(Str)="<<sizeof(Str)<<endl;
    return 0;
}
```

Программа ишлаши натижасида экранга

```
strlen(Str)=10
sizeof(Str)=11
```

хабарлари чиқади.

Одатда *sizeof()* функциясидан *getline()* функциясининг иккинчи аргументи сифати ишлатилади ва сатр узунлигини яққол кўрсат-маслик имконини беради:

```
cin.getline(Satr, sizeof(Satr));
```

**Масала.** Фақат лотин ҳарфларидан ташкил топган сатр берилган. Ундаги ҳар хил ҳарфлар миқдори аниқлансин (ҳарф регистри инобатга олинмасин).

```

int main()
{
    const int n=80;
    char Satr[n];
    cout<<"Satrni kiriting:";
    cin.getline(Satr,sizeof(Satr));
    float s=0;
    int k;
    for(int i=0;i<strlen(Satr);i++)
        if(Satr[i]!=' ')
        {
            k=0;
            for(int j=0;j<strlen(Satr);j++)
                if(Satr[i]==Satr[j]||abs(Satr[i]-
Satr[j])==32)
                    k++;
            s+=1./k;
        }
    cout<<"Satrdagi      turli      harflar      miqdori:
"<<(int)s;
    return 0;
}

```

Программада сатр учун 80 узунлигидаги *Satr* белгилар массиви эълон қилинган ва унинг қиймати клавиатурадан киритилади. Масала қуйидагича ечилади. Ичма-ич жойлашган такрорлаш оператори ёрда-мида *Satr* массивининг ҳар бир элементи - *Satr[i]* массивнинг барча элементлари - *Satr[j]* билан устма-уст тушиши ёки улар бир-биридан 32 сонига фарқ қилиши (ҳарфнинг катта ва кичик регистрдаги кўринишларига мос кодлар ўртасидаги фарқ) ҳолатлари *k* ўзгарувчи-сида саналади ва *s* умумий йиғиндига  $1/k$  қиймати билан қўшилади. Программа охирида *s* қиймати бутун турга айлантирилган ҳолда чоп этилади. Сатрдаги сўзларни бир-биридан ажратувчи пробел белгиси чеклаб ўтилади.

Программанинг сатр киритиш сўровига

**Satrdagi turli harflar miqdori**

сатри киритилса, жавоб тариқасида

**Satrdagi turli belgilar miqdori: 13**

сатри экранга чоп этилади.

### Сатрларни нусхалаш

Сатр қийматини биридан иккинчисига нусхалаш мумкин. Бу мақсадда бир қатор стандарт функциялар аниқланган бўлиб, уларнинг айримларининг тавсифларини келтирамыз.

*strcpy()* функцияси прототипи

**char\*strcpy(char\*str1,const char\*str2)**

кўринишга эга ва бу функция *str2* сатрдаги белгиларни *str1* сатрга байтма-байт нусхалайди. Нусхалаш *str2* кўрсатиб турган сатрдаги нол-терминал учрагунча давом этади. Шу сабабли, *str2* сатр узунлиги *str1* сатр узунлигидан катта эмаслигига ишонч ҳосил қилиш керак, акс ҳолда берилган соҳасида (сегментда) *str1* сатрдан кейин жойлашган берилганлар «устига» *str2* сатрнинг «ортиб қолган» қисми ёзилиши мумкин.

Навбатдаги программа қисми “Satrni nusxalash!” сатрини *Str* сатрга нусхалайди:

```
char Str[20];  
strcpy(Str,"Satrni nusxalash!");
```

Зарур бўлганда сатрнинг қайсидир жойидан бошлаб, охиригача нусхалаш мумкин. Масалан, “Satrni nusxalash!” сатрини 8-белгисидан бошлаб нусха олиш зарур бўлса, уни қуйидагича ечиш мумкин:

```
#include <iostream.h>  
#include <string.h>  
int main()  
{  
    char Str1[20]="Satrni nusxalash!",Str2[20];  
    char* kursatkich=Str1;  
    kursatkich+=7;  
    strcpy(Str2,kursatkich);  
    cout<<Str2<<endl;  
    return 0;  
}
```

*strncpy()* функциясининг *strcpy()* функциясидан фарқли жойи шундаки, унда бир сатрдан иккинчисига нусхаланадиган белгилар сони кўрсатилади. Унинг прототипи қуйидаги кўринишга эга:

**char\*strncpy(char\*str1,const char\*str2,size\_t num);**

Агар *str1* сатр узунлиги *str2* сатр узунлигидан кичик бўлса, ортиқча белгилар «кесиб» ташланади. *strncpy()* функцияси ишлати-лишига мисол кўрайлик:

```
#include <iostream.h>  
#include <string.h>  
int main()  
{  
    char Uzun_str[]="01234567890123456789";  
    char Qisqa_str[]="ABCDEF";  
    strncpy(Qisqa_str,Uzun_str,4);  
    cout <<"Uzun_str= "<<Uzun_str<<endl;  
    cout<<"Qisqa_str="<<Qisqa_str<<endl;
```

```
    return 0;
}
```

Программада *Uzun\_str* сатри бошидан 4 белги *Qisqa\_str* сатрига, унинг олдинги қийматлари устига жойланади ва натижада экранга

```
01234567890123456789
0123EF
```

сатрлар чоп этилади.

*strdup()* функциясига ягона параметр сифатида сатр-манбага кўрсаткич узатилади. Функция, сатрга мос хотирадан жой ажратади, унга сатрни нусхалайди ва юзага келган сатр-нусха адресини жавоб сифатида қайтаради. *strdup()* функция синтаксиси:

```
char*strdup(const char*sourse)
```

Қуйидаги программа бўлагида *satr1* сатрининг нусхаси хотиранинг *satr2* кўрсатган жойида пайдо бўлади:

```
char*satr1="Satr nusxasini olish.";
char*satr2;
satr2=strdup(satr1);
```

### Сатрларни улаш

Сатрларни улаш (конкатенация) амали янги сатрларни ҳосил қилишда кенг қўлланилади. Бу мақсадда «string.h» кутубхонасида *strcat()* ва *strncat()* функциялари аниқланган.

*strcat()* функцияси синтаксиси қуйидаги кўринишга эга:

```
char*strcat(char*str1,const char*str2)
```

Функция ишлаши натижасида *str2* сатр, функция қайтарувчи сатр - *str1* сатр охирига уланади. Функцияни чақиришдан олдин *str1* сатр узунлиги, унга *str2* сатри уланиши учун етарли бўлиши ҳисобга олинган бўлиши керак.

Қуйида келтирилган амаллар кетма-кетлигининг бажарилиши натижасида *satr* сатрига қўшимча сатр уланиши кўрсатилган:

```
char satr[80];
strcpy(satr,"Bu satrga ");
strcat(satr,"satr osti ulandi.");
```

Амаллар кетма-кетлигини бажарилиши натижасида *satr* кўрсатилган жойда "Bu satrga satr osti ulandi." сатри пайдо бўлади.

*strncat()* функцияси *strcat()* функциядан фарқли равишда *str1* сатрга *str2* сатрнинг кўрсатилган узунликдаги сатр қисмини улайди. Уланадиган сатр қисми узунлиги функциянинг учинчи параметри сифатида берилади.

Функция синтаксиси

```
char*strncat(char*str1,const char*str2,size_t
num)
```



Пастда келтирилган программа бўлагида *str1* сатрга *str2* сатр-нинг бошланғич 10 та белгидан иборат сатр қисмини улайди:

```
char satr1[80]="Programmash tillariga misol bu-";
char satr2[80]="C++,Pascal,Basic";
strncat(satr1,satr2,10);
cout<<satr1;
```

Амаллар бажарилиши натижасида экранга

**Programmash tillariga misol bu-C++,Pascal**

сатри чоп этилади.

**Масала.** Нол-терминатор билан тугайдиган *S,S1* ва *S2* сатрлар берилган. *S* сатрдаги *S1* сатр остилари *S2* сатр ости билан алмашти-рилсин. Масалани ечиш учун қуйидаги масала остиларини ечиш зарур бўлади:

3. *S* сатрида *S1* сатр остини кириш ўрнини аниқлаш;
4. *S* сатридан *S1* сатр остини ўчириш;
5. *S* сатрида *S1* сатр ости ўрнига *S2* сатр остини ўрнатиш.

Гарчи бу масала остиларининг ечимлари C++ тилнинг стандарт кутубхоналарида функциялар кўринишида мавжуд бўлса ҳам, улар кодини қайта ёзиш фойдаланувчига бу амалларнинг ички моҳиятини тушунишга имкон беради. Қуйида масала ечимининг программа матни келтирилган:

```
#include <iostream.h>
#include <string.h>
const int n=80;
int Izlash(char*,char*);
void Qirqish(char*,int,int);
void Joylash(char*,char*,int);
int main()
{
    char Satr[n],Satr1[n],Satr2[n];
    cout<<"Satrni kiriting: ";
    cin.getline(Satr,n);
    cout<<"Almashtiriladigan satr ostini kiriting: ";
    cin.getline(Satr1,n);
    cout<<Satr1<<"Qo'yiladigan satrni kiriting: ";
    cin.getline(Satr2,n);
    int Satr1_uzunligi=strlen(Satr1);
    int Satr_osti_joyi;
    do
    {
        Satr_osti_joyi=Izlash(Satr,Satr1);
        if(Satr_osti_joyi!=-1)
        {
```

```

        Qirqish(Satr,Satr_osti_joyi,Satr1_uzunligi);
        Joylash(Satr,Satr2,Satr_osti_joyi);
    }
}
    while (Satr_osti_joyi!=-1);
    cout<<"Almashtirish natijasi: "<<Satr;
    return 0;
}
int Izlash(char satr[],char satr_osti[])
{
    int satr_farqi=strlen(satr)-strlen(satr_osti);
    if(satr_farqi>=0)
    {
        for(int i=0; i<=satr_farqi; i++)
        {
            bool ustma_ust=true;
            for(int j=0;satr_osti[j]!='\0' &&ustma_ust;j++)
                if(satr[i+j]!=satr_osti[j])ustma_ust=false;
            if(ustma_ust)return i;
        }
    }
    return -1;
}

void      Qirqish(char      satr[],int      joy,int
qirqish_soni)
{
    int satr_uzunligi=strlen(satr);
    if (joy<satr_uzunligi)
    {

if(joy+qirqish_soni>=satr_uzunligi)satr[joy]='\0';
        else
            for (int i=0;satr[joy+i-1]!='\0';i++)
                satr[joy+i]=satr[joy+qirqish_soni+i];
    }
}
void      Joylash(char      satr[],char      satr_osti[],int
joy)
{
    char vaqtincha[n];
    strcpy(vaqtincha,satr+joy);
    satr[joy]='\0';
    strcat(satr,satr_osti);
    strcat(satr,vaqtincha);
}

```

}

Программада ҳар бир масала остига мос функциялар тузилган:

1) *int Izlash(char satr[],char satr\_osti[])* - функцияси *satr* сатрига *satr\_osti* сатрининг чап томондан биринчи киришининг ўрнини қайта-ради. Агар *satr* сатрида *satr\_osti* учрамаса -1 қийматини қайтаради.

2) *void Qirqish(char satr[],int joy,int qirqish\_soni)* - функцияси *satr* сатрининг *joy* ўрнидан бошлаб *qirqish\_soni* сондаги белгиларни қирқиб ташлайди. Функция натижаси *satr* сатрида ҳосил бўлади;

3) *void Joylash(char satr[],char satr\_osti[],int joy)* - функцияси *satr* сатрига, унинг *joy* ўрнидан бошлаб *satr\_osti* сатрини жойлаштиради.

Бош функцияда сатр (*S*), унда алмаштириладиган сатр (*S1*) ва *S1* ўрнига жойлаштириладиган сатр (*S2*) оқимдан ўқилади. Такрорлаш оператори бажарилишининг ҳар бир қадамида *S* сатрининг чап томо-нидан бошлаб *S1* сатри изланади. Агар *S* сатрида *S1* мавжуд бўлса, у қирқилади ва шу ўринга *S2* сатри жойлаштирилади. Такрорлаш жараёни *Izlash()* функцияси -1 қийматини қайтаргунча давом этади.

### Сатрларни солиштириш

Сатрларни солиштириш улардаги мос ўринда жойлашган белгилар кодларини ўзаро солиштириш билан аниқланади. Бунинг учун «string.h» кутубхонасида стандарт функциялар мавжуд.

*strcmp()* функцияси синтаксиси

***int strcmp(const char\*str1, const char\*str2)***

кўринишига эга бўлиб, функция *str1* ва *str2* солиштириш натижаси сифатида сон қийматни қайтаради (масалан, бутун *i* ўзгарувчисиди) ва улар куйидагича изоҳланади:

1) *i<0* - агар *str1* сатри *str2* сатридан кичик бўлса;

2) *i=0* - агар *str1* сатри *str2* сатрига тенг бўлса;

3) *i>0* - агар *str1* сатри *str2* сатридан катта бўлса.

Функция ҳарфларнинг регистрини фарқлайди. Буни мисолда кўришимиз мумкин:

```
char                                satr1[80]="Programmalash
tillari:C++,pascal.";
char                                satr2[80]="Programmalash
tillari:C++,Pascal.";
int i;
i=strcmp(satr1,satr2);
```

Натижада *i* ўзгарувчиси мусбат қиймат қабул қилади, чунки солиштирилаётган сатрлардаги «*pascal*» ва «*Pascal*» сатр қисмларида биринчи ҳарфлар фарқ қилади. Келтирилган мисолда *i* қиймати 32 бўлади. Бу фарқланувчи ҳарфлар кодларининг айирмаси. Агар функцияга

```
i= strcmp(satr2,satr1);
```

кўринишида мурожаат қилинса *i* қиймати манфий сон -32 бўлади.

Агар сатрлардаги бош ёки кичик ҳарфларни фарқламасдан солиштириш амалини бажариш зарур бўлса, бунинг учун *strcmpi()* функциясидан фойдаланиш мумкин. Юқорида келтирилган мисол-даги сатрлар учун

```
i=strcmpi(satr2,satr1);
```

амали бажарилганда *i* қиймати 0 бўлади.

*strncmp()* функцияси синтаксиси

```
int strncmp(const char*str1,const char*str2,size_t num);
```

кўринишида бўлиб, *str1* ва *str2* сатрларни бошланғич *num* сонидagi белгиларини солиштиради. Функция ҳарфлар регистрини инобатга олади. Юқорида мисолда аниқланган *satr1* ва *satr2* сатрлар учун

```
i=strncmp(satr1,satr2,31);
```

амали бажарилишида *i* қиймати 0 бўлади, чунки сатрлар бошидаги 31 белгилар бир хил.

*strncmpi()* функцияси *strncmp()* функциясидек амал қилади, фарқли томони шундаки, солиштиришда ҳарфларнинг регистрини ҳисобга олинмайди. Худди шу сатрлар учун

```
i=strncmpi(satr1,satr2,32);
```

амали бажарилиши натижасида *i* ўзгарувчи қиймати 0 бўлади.

### Сатрдаги ҳарфлар регистрини алмаштириш

Берилган сатрдаги кичик ҳарфларни бош ҳарфларга ёки тескари-сига алмаштиришга мос равишда *\_strupr()* ва *\_strlwr()* функциялар ёрдамида амалга ошириш мумкин. Компиляторларнинг айрим вариантларида функциялар номидаги тагчилик ('\_') бўлмаслиги мумкин.

*\_strlwr()* функцияси синтаксиси

```
char* _strlwr(char* str);
```

кўринишида бўлиб, аргумент сифатида берилган сатрдаги бош ҳарф-ларни кичик ҳарфларга алмаштиради ва ҳосил бўлган сатр адресини функция натижаси сифатида қайтаради. Қуйидаги программа бўлаги *\_strlwr()* функциясидан фойдаланишга мисол бўлади.

```
char str[]="10 TA KATTA HARFLAR";  
_strlwr(str);  
cout<<str;
```

Натижада экранга

```
10 ta katta harflar
```

сатри чоп этилади.

`_strupr()` функцияси худди `_strlwr()` функциясидек амал қилади, лекин сатрдаги кичик ҳарфларни бош ҳарфларга алмаштиради:

```
char str[]="10 ta katta harflar";
strupr(str);
cout<<str;
```

Натижада экранга

**10 TA KATTA HARFLAR**

сатри чоп этилади.

Программалаш амалиётида белгиларни қайсидир ораликқа тегишли эканлигини билиш зарур бўлади. Буни «ctype.h» сарлавҳа файлида эълон қилинган функциялар ёрдамида аниқлаш мумкин. Қуйида уларнинг бир қисмининг тавсифи келтирилган:

`isalnum()` - белги рақам ёки ҳарф (*true*) ёки йўқлигини (*false*) аниқлайди;

`isalpha()` - белгини ҳарф (*true*) ёки йўқлигини (*false*) аниқлайди;

`isascii()` - белгини коди 0..127 оралиғида (*true*) ёки йўқлигини (*false*) аниқлайди;

`isdigit()` - белгини рақамлар диапазонида тегишли (*true*) ёки йўқлигини (*false*) аниқлайди.

Бу функциялардан фойдаланишга мисол келтирамиз.

```
#include <iostream.h>
#include <ctype.h>
#include <string.h>
int main()
{
    char satr[5];
    int xato;
    do
    {
        xato=0;
        cout<<"\nTug'ilgan yilingizni kiriting: ";
        cin.getline(satr,5);
        for (int i=0; i<strlen(satr) && !xato; i++)
        {
            if(isalpha(satr[i]))
            {
                cout<<"Harf kiritdildi!";
                xato=1;
            }
            else
                if(iscntrl(satr[i]))
                {
                    cout<<"Boshqaruv belgisi kiritildi!";
```

```

        xato=1;
    }
    else
        if (ispunct(satr[i]))
        {
            cout<<"Punktuatsiya belgisi kiritildi!";
            xato=1;
        }
        else
            if (!isdigit(satr[i]))
            {
                cout<<"Raqamdan          farqli          belgi
kiritdildi!";
                xato=1;
            }
    }
    if (!xato)
    {
        cout<<"Sizni tug'ilgan yilingiz: "<<satr;
        return 0;
    }
} while(1);
}

```

Программада фойдаланувчига туғилган йилини киритиш таклиф этилади. Киритилган сана *satr* ўзгарувчисига ўқилади ва агар сатрнинг ҳар бир белгиси (*satr[i]*) ҳарф ёки бошқарув белгиси ёки пунктуация белгиси бўлса, шу ҳақда хабар берилади ва туғилган йилни қайта киритиш таклиф этилади. Программа туғилган йил (тўртта рақам) тўғри киритилганда “Sizni tug'ilgan yilingiz: XXXX” сатрини чоп қилиш билан ўз ишини тугатади.

### Сатрни тескари тартиблаш

Сатрни тескари тартиблашни учун *strrev()* функциясидан фойдаланиш мумкин. Бу функция қуйидагича прототипга эга:

```
char*strrev(char*str) ;
```

Сатр реверсини ҳосил этишга мисол:

```
char str[]="telefon";
cout<<strrev(str) ;
```

амаллар бажарилиши натижасида экранга

```
nofelet
```

сатри чоп этилади.

## Сатрда белгини излаш функциялари

Сатрлар билан ишлашда ундаги бирорта белгини излаш учун «string.h» кутубхонасида бир қатор стандарт функциялар мавжуд.

Бирорта белгини берилган сатрда бор ёки йўқлигини аниқлаб берувчи *strchr()* функциясининг прототипи

```
char*strchr(const char*str,int c);
```

Кўринишида бўлиб, у *c* белгинининг *str* сатрида излайди. Агар излаш мувофақиятли бўлса, функция шу белгининг сатрдаги ўрнини (адресини) функция натижаси сифатида қайтаради, акс ҳолда, яъни белги сатрда учрамаса функция *NULL* қийматини қайтаради. Белгини излаш сатр бошидан бошланади.

Қуйида келтирилган программа бўлаги белгини сатрдан излаш билан боғлиқ.

```
char satr[]="0123456789";  
char*pSatr;  
pSatr=strchr(satr,'6');
```

Программа ишлаши натижасида *pSatr* кўрсаткичи *satr* сатрининг '6' белгиси жойлашган ўрнига мос адресни кўрсатади.

*strrchr()* функцияси берилган белгини берилган сатр охиридан бошлаб излайди. Агар излаш муваффақиятли бўлса, белгини сатрга охирги киришининг ўрнини қайтаради, акс ҳолда *NULL*.

Мисол учун

```
char satr[]="0123456789101112";  
char*pSatr;  
pSatr=strrchr(satr,'0');
```

амалларини бажарилишида *pSatr* кўрсаткичи *satr* сатрининг "01112" сатр қисмининг бошланишига кўрсатади.

*strspn()* функцияси иккита сатрни белгиларни солиштиради. Функция куйидаги

```
size_t strspn(const char*str1, const char*str2);
```

кўринишга эга бўлиб, у *str1* сатрдаги *str2* сатрга кирувчи бирорта белгини излайди ва агар бундай элемент топилса, унинг индекси функция қиймати сифатида қайтарилади, акс ҳолда функция сатр узунлигидан битта ортиқ қийматни қайтаради.

Мисол:

```
char satr1[]="0123ab6789012345678";  
char satr2[]="a32156789012345678";  
int farqli_belgi;  
farqli_belgi=strspn(satr1,satr2);  
cout<<"Satr1 satridagi Satr2 satrga kirmaydigan\  
birinchi belgi indexsi = "<<farqli_belgi;
```

```
cout<<"va u '"<<satr1[farqli_belgi]<<"'
belgisi.";
```

амаллар бажарилиши натижасида экранга

```
Satrlardagi mos tushmagan belgi indexsi = 5
```

сатри чоп этилади.

*strcspn()* функциясининг прототипи

```
size_t strcspn(const char* str1, const char*
str2);
```

кўринишида бўлиб, у *str1* ва *str2* сатрларни солиштиради ва *str1* сатрининг *str2* сатрига кирган биринчи белгини индексини қайтаради. Масалан,

```
char satr[]="Birinchii satr";
int index;
index=strcspn(satr,"sanoq tizimi");
```

амаллари бажарилгандан кейин *index* ўзгарувчиси 1 қийматини қабул қилади, чунки биринчи сатрнинг биринчи ўриндаги белгиси иккинчи сатрда учрайди.

*strpbrk()* функциясининг прототипи

```
char* strpbrk(const char* str1, const char*
str2);
```

кўринишга эга бўлиб, у *str1* сатрдаги *str2* сатрга кирувчи бирорта белгини излайди ва агар бундай элемент топилса, унинг адреси функция қиймати сифатида қайтарилади, акс ҳолда функция *NULL* қиймати қайтаради. Қуйидаги мисол функцияни қандай ишлашини кўрсатади.

```
char satr1[]="0123456789ABCDEF";
char satr2[]="ZXYabcdefABC";
char* element;
element = strpbrk(satr1,satr2);
cout<<element<<' \n' ;
```

Программа ишлаши натижасида экранга *str1* сатрининг

```
ABCDEF
```

сатр остиси чоп этилади.

### Сатр қисмларини излаш функциялари

Сатрлар билан ишлашда бир сатрда иккинчи бир сатрнинг (ёки унинг бирор қисмини) тўлиқ киришини аниқлаш билан боғлиқ масала-лар нисбатан кўп учрайди. Масалан, матн таҳрирларидаги сатрдаги бирорта сатр қисмини иккинчи сатр қисми билан алмаштириш масаласини мисол келтириш мумкин (юқорида худди шундай масала учун программа келтирилган). Стандарт «string.h» кутубхонаси бу тоифадаги масалалар учун қуйидаги функциялар мавжуд.



*strstr()* функцияси қуйидагича эълон қилинади:

```
char*strstr(const char*str,const char*substr);
```

Бу функция *str* сатрига *substr* сатр қисми кириши текширади, агар *substr* сатр қисми *str* сатрига тўлиқ кириши мавжуд бўлса, сатрнинг чап томонидан биринчи киришдаги биринчи белгининг адреси жавоб тариқасида қайтарилади, акс ҳолда функция *NULL* қийматини қайтаради.

Қуйидаги мисол *strstr()* функциясини ишлатишни кўрсатади.

```
char satr1[]=  
"Satrdan satr ostisi izlanmoqda, satr ostisi  
mavjud";  
char satr2[]="satr ostisi";  
char* satr_osti;  
satr_osti=strstr(satr1,satr2);  
cout<<satr_osti<<'\\n' ;
```

Программа буйруқлари бажарилиши натижасида экранга

```
satr ostisi izlanmoqda, satr ostisi mavjud
```

сатри чоп этилади.

Кейинги программа бўлагиди сатрда бошқа бир сатр қисми мавжуд ёки йўқлигини назорат қилиш ҳолати кўрсатилган:

```
char Ismlar[]=  
"Alisher,Farxod, Munisa, Erkin, Akmal, Nodira";  
char Ism[10];  
char*Satrdagi_ism;  
cout<<"Ismni kiriting: ";  
cin>>Ism;  
Satrdagi_ism=strstr(Ismlar,Ism);  
cout<<"Bunaqa ism ru'yxatda ";  
if(Satrdagi_ism==NULL)cout<<"yo'q.\\n";  
else cout<<"bor.\\n";
```

Программада фойдаланувчидан сатр қисми сифатида бирорта номни киритиш талаб қилинади ва бу қиймат *Ism* сатрига ўқилади. Киритилган исм программада аниқланган рўйхатда (*Ismlar* сатрида) бор ёки йўқлиги аниқланади ва хабар берилади.

*strtok()* функциясининг синтаксиси

```
char* strtok(char* str, const char* delim);
```

кўринишда бўлиб, у *str* сатрида *delim* сатр-рўйхатида берилган ажратувчилар оралиғига олинган сатр қисмларни ажратиб олиш имконини беради. Функция биринчи сатрда иккинчи сатр-рўйхатдаги ажратув-чини учратса, ундан кейин нол-терминаторни қўйиш орқали *str* сатрни иккига ажратади. Сатрнинг иккинчи бўлагидан ажратувчилар билан «ўраб олинган» сатр қисмлари топиш учун функцияни кейинги чақирилишида

биринчи параметр ўрнига *NULL* қийматини қўйиш керак бўлади. Қуйидаги мисолда сатрни бўлакларга ажратиш масаласи қаралган:

```
#include <iostream.h>
#include <string.h>
int main()
{
    char Ismlar[]=
    "Alisher,Farxod Munisa, Erkin? Akmal0, Nodira";
    char Ajratuvchi[]=" ,!?.0123456789";
    char*Satrdagi_ism;
    Satrdagi_ism=strtok(Ismlar,Ajratuvchi);
    if(Satrdagi_ism) cout<<Satrdagi_ism<<'\\n';
    while(Satrdagi_ism)
    {
        Satrdagi_ism=strtok(NULL,Ajratuvchi);
        if(Satrdagi_ism) cout<<Satrdagi_ism<<'\\n';
    }
    return 0;
}
```

Программа ишлаши натижасида экранга *Ismlar* сатридаги ‘ ’ (пробел), ‘,’ (вергул), ‘?’ (сўроқ белгиси) ва ‘0’ (рақам) билан ажра-тилган сатр қисмлари - исмлар чоп қилинади:

```
Alisher
Farxod
Munisa
Erkin
Akmal
Nodira
```

### 34-мавзу: Тузилмалар.

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

- 1.С++ тилидаги программа тузилиши ва унинг компиляцияси
- 2.С++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

### **Tuzilmalar va Birlashmalar**

Ma'lumki, biror predmet sohasidagi masalani echishda undagi ob'ektlar bir nechta, har xil turdagi parametrlar bilan aniqlanishi mumkin. Masalan tekislikdagi nuqta haqiqiy turdagi x-absissa y-ordinata juftligi (x,y) ko'rinishda beriladi. Talaba haqidagi ma'lumotlar –sitr turidagi talaba familiya, ismi sharifi mutaxassislik yo'nalish, talaba yashash manzili, butun turdagi tug'ilgan yili, o'quv bosqichi, haqiqiy turdagi reyting bali, mantiqiy turdagi talaba jinsi haqidagi ma'lumotlar va boshqa ma'lumotlardan shakllanadi.

Programmada holat yoki tushuncha tafsiflovchi har bir berilganlar uchun alohida o'zgaruvchi aniqlab masalani yechish mumkin. Lekin bu holda obyekt haqidagi malumotlar <<tarqoq>> bo'ladi, ularni qayta ishlash murakkablashadi, obyekt haqidagi berilganlar jamlangan holda ko'rish qiyinlashadi.

C++ tilida bir yoki bir xil turdagi berilganlar jamlanmasi tuzilma deb nomlanadi. Tuzilma foydalanuvchi tomonidan aniqlangan berilganlarning yangi turi hisoblanadi. Tuzilma quidagicha aniqlanadi:

```
struct <tuzilma ismi>
{
    <tur_1> <nom_1> ;
    <tur_2> <nom_2> ;
    .....
    <tur_n> <nom_n> ;
};
```

Bu erda <tuzilma nomi > tuzilma ko'rinishida yaratilayotgan yangi turning nomi "<tur\_i> <nom\_i> ;"—tuzilmaning i-maydonining (nom\_i) e'loni.

Boshqacha aytganda, tuzilma e'lon qilingan o'zgaruvchilardan (maydonlardan) tashkil topadi. Unga har xil turdagi berilganlarni o'z ichiga oluvchi qobiq deb aqrash mumkin. Qobiqdagi berilganlarni yaxlit holda ko'chirish tashqi qurilmalarga ( turlangan fayllarga ) yozish, o'qish mumkin bo'ladi.

Talaba haqidagi berilganlarni o'z ichiga oluvchi tuzilma turining e'lon qilinishini ko'raylik :

```
struct Talaba
{
    char FISH[30];
    unsigned int tug'_yil;
    unsigned int kurs ;
    char Yo'nalish[20];
    float Reyting ;
    unsigned char Jinsi[5];
    char manzil [40];
    bool status;
};
```

Programmada tuzilmalardan foydalanish shu turdagi o'zgaruvchilar e'lon qilinishi va ularni qayta ishlash orqali amalgam oshiriladi:

Talaba talaba ;

Tuzilma turining e'lonida turning nomi bo'lmasligi mumkin ,lekin bu holda tuzilma aniqlanishidan keyin albatta o'zgaruvchilar nomlari yozilishi kerak:

```
struct Talaba
{
    unsigned int x,y;
    unsigned char Rang ;
    }Point1,Point2;
```

Keltirilgan misolda tuzilma turidagi Point1,Point2 o'zgaruvchilari e'lon qilingan.

Tuzilma turidagi o'zgaruvchilar bilan ishlash ,uning maydonlari bilan ishlashni anglatadi.Tuzilma maydoniga murojat qilish '.' (nuqta) orqali amalgam oshiriladi .Bunda tuzilma turidagi o'zgaruvchi nomi ,undan keyin huqta qo'yiladi va o'zgaruvchining nomi yoziladi.Masalan,talaba haqidagi tuzilma maydonlariga murojat quidagicha bo'ladi:

```
talaba .kurs=2;
talaba tug'_yil=1988;
strcpy(talaba.Yo'nalish,"Amaliy matematika va Informatika");
strcpy(talaba.Jinsi,"Erkak");
strcpy(talaba.Manzil,"Toshkent,Yunusobod 6-3-8,tel:224-45-78");
talaba.Reyting =123.52;keltirilgan misolda talaba tuzilmasining son turidagi
```

maydonlariga oddiy ko'rinishda qiymatlar berilgan ,sitr turidagi maydonlar uchun strcpy funksiyasi orqali qiymat berish amalgam oshirilgan.

Tuzilma turidagi ob'ektning xotiradan qancha joy egallanganmligini sizeof funksiyasi (operatori) orqali aniqlash mumkin .

```
int i=sizeof(Talaba);
```

ayrim holatlarda tuzilma maydomlari o'lchamini bitlarda aniqlash orqali egallanadigan xotirani qisqartirish mumkin .Buning uchun tuzilma maydoni quidagicha e'lon qilinadi :

<maydon nomi>:<o'zgarmas ifoda >

Bu erda <maydon nomi>-maydon turi va nomi <o'zgarmas ifoda >-maydonning bitlardagi uzunligi.Maydon turi butun turlar bo'lishi kerak (int,long ,unsigned,char).

Agar foydalanuvchi tuzilmaning maydoni faqat 0 va 1 qiymatni qabul qilishini bilsa ,bu maydon uchun bir bit joy ajratishi mumkin (bir bayt yoki ikki bayt o'rniga ).Xotirani tejash evaziga maydon ustida amal bajarishda razryadli arifmetikani qo'llash zarur bo'ladi.

Misol uchun sana-vaqt bilan bog'liq tuzilmani yaratishning ikkita variantini ko'raylik.Tuzilma Yil,Oy,Kun,soat,minut va sekund maydonlaridan iborat bo'lsin va uni quidagicha aniqlash mumkin

```
struct Sana_vaqt
{
    unsigned short Yil;
    unsigned short Oy;
    unsigned short Kun;
```

```

unsigned short Soat;
unsigned short Minut;
unsigned short Sekund;
};

```

Bunday aniqlashda Sana\_vaqt tuzilmasi xotirada 6 maydon \*2 bayt=12bayt joy egallaydi. Agar e'tibor berilsa tuzilmada ortiqcha joy egallash holatlari mavjud. Masalan ,yil qiymati 0 dan 99 gacha qiymat bilan aniqlanishi etarli (masalan ,2008 yilni 8 qiymati bilan ifodalash mumkin). Shuning uchun unga 2 bayt emas , balki 7 bit ajratish etatish yetarli. Huddi shunday oy uchun 1...12 qiymatlari ifodalashga yetarli 4 bit joy etarli va hakazo.

Yuqoridagi keltirilgan cheklovlardan keyin sana-vaqt tuzilmasini tejamkor variantini aniqlash mumkin:

```

struct Sana_vaqt2
{
    unsigned Yil:7;
    unsigned Oy:4;
    unsigned Kun:5;
    unsigned Soat:6;
    unsigned Minut:6;
    unsigned Sekund:6;
};

```

Bu tuzilma xotiradan 5 bayt joy egallaydi.

### **Tuzilma Funksiya argumenti sifatida**

Tuzimalar funksiya argumenti sifatida ishlatilishi mumkin . Buning uchun Funksiya protatibifa foydalanuvchi tomonidan tuzilmani e'lon qilishda aniqlangan berilganlar turi ko'rsatilishi kerak bo'ladi. Masalan talaba haqida berilganlarni o'z ichiga oluvchi Talaba tuzilmasi turidagi berilganlarni Talaba\_Manzili() funksiyasiga parametr sifatida berish uchun funksiya protatipi quifagi ko'rinishda bo'lishi kerak :

```

void Talaba_Manzili(Talaba );

```

funksiyaga tuzilmani argument sifatida uzatishga misol sifatida programma matni :

```

#include <iostream.h>
#include <conio.h>
#include <string .h>
struct Talaba
{
    char FISH[30];
    unsigned int Tug' _yil;
    unsigned int Kurs;
    char Yo'nalish[50];
    float Reyting ;
    unsigned char Jinsi[5];

```

```

char Manzil[50];
bool status;
};
Void Talaba_Manzili(Talaba);
int main(int argc, char* argv[])
{
    Talaba talaba ;
    talaba.Kurs=2;
    talaba.Tug'_yil=1988;
    strcpy(talaba.FISh,"Abdullayev.A.A");
    strcpy(talaba.Yo'nalish,"Amaliy matematika va Informatika");
    strcpy(talaba.Jinsi,"Erkak");
    strcpy(talaba.Manzil,"Toshkent,Yunusobod 6-3-8,tel:224-45-78");
    talaba.Reyting=123.52;
    Talaba_Manzili(talaba);
    Return 0;
}
void Talaba_Manzili(talaba t)
{
    cout<<"Talaba.FISh:"<<t. FISh<<endl;
    cout<<"Manzili:"<<t. Manzil<<endl;
}

```

Programma bosh funksiyasida talaba tuzilmasi aniqlanib ,uning maydonlariga qiymatlar beriladi.Keyin talaba tuzilmasi Talaba\_Manzili() funksiyasiga argument sifatida uzatiladi.Programma ishlashi natijasida ekranga quidagi ma'lumotlar chop etiladi.

Talaba FISh:Abdullayeva A.A.

Manzili : Toshkent,Yunusobod 6-3-8,tel:224-45-78

### Tuzilmalar massivi

O'z-o'zidan malumki, bitta tuzilma turidagi qiymat bilan yechish mumkin bo'lgan masalalar doirasi juda tor va aksariyat holatlarda qo'yilgan masala tuzilmalar majmuasi bilan ishlashni talab qiladi.

### 35-мавзу: Бирлашмалар.

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

- 1.C++ тилидаги программа тузилиши ва унинг компиляцияси
- 2.C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

### Бирлашмалар ва улар устида амаллар

Бирлашмалар хотиранинг битта соҳасида (битта адрес бўйича) ҳар хил турдаги бир нечта берилганларни сақлаш имконини беради.

Бирлашма эълони *union* калит сўзи, ундан кейин идентификатор ва блок ичида ҳар хил турдаги элементлар эълонидан иборат бўлади, масалан:

```
union Birlashma
{
  int n;
  unsigned long N;
  char Satr[10];
};
```

Бирлашманинг бу эълонида компилятор томонидан *Birlashma* учун унинг ичидаги энг кўп жой эгалловчи элементнинг - *Satr* сатрининг ўлчамида, яъни 10 байт жой ажратилади. Вақтнинг ҳар бир моментида бирлашмада, эълон қилинган майдонларнинг фақат биттасининг туридаги берилган мавжуд деб ҳисобланади. Юқоридаги мисол-да *Birlashma* устида амал бажарилишида унинг учун ажратилган хотирада ёки *int* туридаги *n* ёки *unsigned long* туридаги *N* ёки *Satr* сатр қиймати жойлашган деб ҳисобланади.

Бирлашма майдонларига худди структура майдонларига муро-жаат қилгандек ‘.’ орқали мурожаат қилинади.

Структуралардан фарқли равишда бирлашма эълонида фақат унинг биринчи элементига бошланғич қиймат бериш мумкин:

```
union Birlashma
{
  int n;
  unsigned long N;
  char Satr[10];
}
birlashma={25};
```

Бу мисолда *birlashma* бирлашмасининг *n* майдони бошланғич қиймат олган ҳисобланади.

Бирлашма элементи сифатида структуралар келиши мумкин ва улар одатда берилганни «бўлакларга» ажратиш ёки «бўлаклардан» яхлит берилганни ҳосил қилиш учун хизмат қилади. Мисол учун сўзни байтларга, байтларни тетрадаларга (4 разрядга) ажратиш ва қайтадан бирлаштириш мумкин.

Қуйида байтни катта ва кичик ярим байтларга ажратишда бирлашма ва структурадан фойдаланилган программани матни келтирилган.

```

#include <iostream.h>
union BCD
{
    unsigned char bayt;
    struct
    {
        unsigned char lo:4;
        unsigned char hi:4;
    } bin;
} bcd;
int main()
{
    bcd.bayt=127;
    cout<<"\nKatta yarim bayt : "<<(int)bcd.bin.hi;
    cout<<"\nKichik yarim bayt: "<<(int)bcd.bin.lo;
    return 0;
}

```

Программа бош функциясида *bcd* бирлашмасининг байт ўлчами-да *bayt* майдонига 127 қиймати берилади ва унинг катта ва кичик ярим байтлари чоп этилади.

Программа ишлаши натижасида экранга қуйидаги натижалар чиқади:

```

Katta yarim bayt : 7
Kichik yarim bayt: 15

```

**Масала.** Ҳақиқий турдаги соннинг компьютер хотирасидаги ички кўринишини чоп қилиш. Ҳақиқий сон *float* турида деб ҳисоб-ланади ва у хотирада 4 байт жой эгаллайди (1-иловага қаранг). Қўйилган масалани ечиш учун бирлашма хусусиятдан фойдаланилади, яъни хотиранинг битта адресига ҳақиқий сон ва белгилар массиви жойлаштирилади. Ҳақиқий сон хотирага ўқилиб, белгилар массивининг ҳар бир элементининг (байтининг) иккилик кўриниши чоп этилади.

Программа матни:

```

#include <iostream.h>
const unsigned char bitlar_soni=7;
const unsigned char format=sizeof(float);
void Belgi_2_kodi(unsigned char blg);
union Son_va_Belgi
{
    float son;
    unsigned char belgi[format];
};
int main()
{

```



```

Son_va_Belgi son_va_belgi;
cin>>son_va_belgi.son;
for(int b=format-1; b>=0; b--)
    Belgi_2_kodi(son_va_belgi.belgi[b]);
return 0;
}
void Belgi_2_kodi(unsigned char blg)
{
    unsigned char l00000000=128;
    for(int i=0;i<=bitlar_soni;i++)
    {
        if(blg&l00000000)cout<<'1';
        else cout<<'0';
        blg=blg<<1;
    }
    cout<<' ';
}

```

Программада *Son\_va\_Belgi* бирлашмасини эълон қилиш орқали *float* туридаги *x* ўзгарувчисини ва *float* тури форматининг байтлардаги узунлигидаги белгилардан иборат *belgi* массивини хотиранинг битта жойига жойлашувига эришилади. Бош функцияда бирлашма туридаги *son\_va\_belgi* ўзгарувчиси эълон қилинади ва унинг *x* майдонига клавиатурадан ҳақиқий сон ўқилади. Кейин белгилар массивидаги ҳар бир элементнинг иккилик коди чоп этилади. Иккилик кодни чоп этиш 8 марта байтни 7-разрядидаги сонни чоп этиш ("*blg&l00000000*" амалининг натижасига кўра) ва байт разрядларини биттага чапга суриш орқали амалга оширилади. Шунга эътибор бериш керакки, белгилар массивидаги элементларнинг иккилик кодларини чоп қилиш ўнгдан чап томонга бажарилган. Бунга сабаб, сон ички форматидаги байтларнинг хотирада «*кичик байт - кичик адресда*» қоидасига кўра жойлашувидир.

Программага -8.5 сони киритилса, экранда

**11000001 00001000 00000000 00000000**

кўринишидаги иккилик сонлари кетма-кетлиги пайдо бўлади.

### 36-мавзу: Дастурни созлаш технологиялари

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

- 1.С++ тилидаги программа тузилиши ва унинг компиляцияси
- 2.С++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

### **37-мавзу: Файл тушунчаси. Матн файллар.**

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

**ДАРС Р Е Ж АСИ (асосий саволлар):**

1. C++ тилидаги программа тузилиши ва унинг компиляцияси
2. C++ тилида бажарилувчи файл яратиш босқичлари

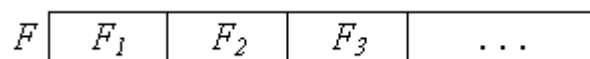
Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

#### **Файл тушунчаси**

C++ тилидаги стандарт ва фойдаланувчи томонидан аниқланган турларнинг муҳим хусусияти шундан иборатки, уларнинг олдиндан аниқланган миқдордаги чекли элементлардан иборатлигидир. Ҳатто берилганлар динамик аниқланганда ҳам, оператив хотиранинг (уюм-нинг) амалда чекланганлиги сабабли, бу берилганлар миқдори юқори-дан чегараланган элементлардан иборат бўлади. Айрим бир тадбиқий масалалар учун олдиндан берилганнинг компоненталари сонини аниқлаш имкони йўқ. Улар масалани ечиш жараёнида аниқланади ва етарлича катта ҳажмда бўлиши мумкин. Иккинчи томондан, програм-мада эълон қилинган ўзгарувчиларнинг қийматлари сифатида аниқ-ланган берилганлар фақат программа ишлаш пайтидагина мавжуд бўлади ва программа ўз ишини тугатгандан кейин йўқолиб кетади. Агар программа янгидан ишга туширилса, бу берилганларни янгидан ҳосил қилиш зарур бўлади. Аксарият тадбиқий масалалар эса берилганларни доимий равишда сақлаб туришни талаб қилади. Масалан, корхона ходимларининг ойлик маошини ҳисобловчи программада ходимлар рўйхатини, штат ставкалари ва ходимлар томонидан олинган маошлар ҳақидаги маълумотларни доимий равишда сақлаб туриш зарур. Бу талабларга файл туридаги объектлар (ўзгарувчилар) жавоб беради.

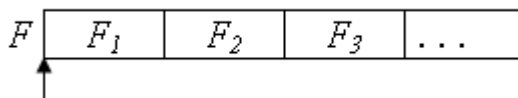
*Файл* - бу бир хил турдаги қийматлар жойлашган ташқи хотирадаги номланган соҳадир.

Файлни, бошида кетма-кет равишда жойлашган ёзувлар (масалан, мусиқа) билан тўлдирилган ва охири бўш бўлган етарлича узун магнит тасмасига ўхшатиш мумкин.



11.1-расм. Файл тасвири

11.1-расмда  $F$ - файл номи,  $F_1, F_2, F_3$  - файл элементлари (компоненталари). Худди янги мусиқани тасма охирига қўшиш мумкин бўлгандек, янги ёзувлар файл охирига қўшилиши мумкин.



11.2-расм. Файл кўрсаткичи

Яна бир муҳим тушунчалардан бири файл кўрсаткичи тушунчасидир. *Файл кўрсаткичи* - айна пайтда файлдан ўқиладиган ёки унга ёзиладиган жой (ёзув ўрни) кўрсатиб туради, яъни файл кўрсаткичи кўрсатиб турган жойдан битта ёзувни ўқиш ёки шу жойга янги ёзувни жойлаштириш мумкин. 11.2-расмда файл кўрсаткичи файл бошини кўрсатмоқда.

Файл ёзувларига мурожаат кетма-кет равишда амалга оширилади:  $n$ - ёзувга мурожаат қилиш учун  $n-1$  ёзувни ўқиш зарур бўлади. Шунинг билан таъкидлаб ўтиш зарурки, файлдан ёзувларни ўқиш жараёни қисман «автоматлашган», унда  $i$  - ёзувни ўқилгандан кейин, кўрсаткич навбатдаги  $i+1$  ёзув бошига кўрсатиб туради ва шу тарзда ўқишни давом эттириш мумкин (массивлардагидек индексни ошириш шарт эмас). Файл - бу берилганларни сақлаш жойидир ва шу сабабли унинг ёзувлари устида тўғридан-тўғри амал бажариб бўлмайди. Файл ёзуви устида амал бажариш учун ёзув қиймати оператив хотирага мос турдаги ўзгарувчига ўқилиши керак. Кейинчалик, зарур амаллар шу ўзгарувчи устида бажарилади ва керак бўлса натижалар яна файлга ёзилиши мумкин.

Операцион система нуқтаи-назаридан файл ҳисобланган ҳар қандай файл C++ тили учун *физик файл* ҳисобланади. MS DOS учун физик файллар <файл номи>.<файл кенгайтмаси> кўринишидаги «8.3» форматидаги сатр (ном) орқали берилади. Файл номлари сатр ўзгармаслар ёки сатр ўзгарувчиларида берилиши мумкин. MS DOS қоидаларига кўра файл номи тўлиқ бўлиши, яъни файл номининг бошида адрес қисми бўлиши мумкин: "C:\\USER\\Misol.cpp", "A:\\matn.txt".

C++ тилида *мантиқий файл* тушунчаси бўлиб, у файл туридаги ўзгарувчини англатади. Файл туридаги ўзгарувчиларга бошқа турдаги ўзгарувчилар каби қиймат бериш оператори орқали қиймат бериб бўлмайди. Бошқача айтганда файл туридаги ўзгарувчилар устида ҳеч қандай амал аниқланмаган. Улар устида бажариладиган барча амаллар функциялар воситасида бажарилади.

Файллар билан ишлаш қуйидаги босқичларни ўз ичига олади:

- файл ўзгарувчиси албатта дискдаги файл билан боғланади;
- файл очилади;

- файл устида ёзиш ёки ўқиш амаллари бажарилади;
- файл ёпилади;
- файл номини ўзгартириш ёки файлни дискдан ўчириш амалларини бажарилиши мумкин.

### Матн ва бинар файллар

C++ тили C тилидан ўқиш-ёзиш амалини бажарувчи стандарт функциялар кутубхонасини ворислик бўйича олган. Бу функциялар “stdio.h” сарлавҳа файлида эълон қилинган. Ўқиш-ёзиш амаллари файллар билан бажарилади. Файл матн ёки бинар (иккилик) бўлиши мумкин.

*Матн файл* - ASCII кодадаги белгилар билан берилганлар мажмуаси. Белгилар кетма-кетлиги сатрларга бўлинган бўлади ва сатр-нинг тугаш аломати сифатида CR (кареткани қайтариш ёки ‘\r’) LF (сатрни ўтказиш ёки ‘\n’) белгилар жуфтлиги ҳисобланади. Матн файлдан берилганларни ўқишда бу белгилар жуфтлиги битта CR белгиси билан алмаштирилади ва аксинча, ёзишда CR белгиси иккита CR ва LF белгиларига алмаштирилади. Файл охири #26 (^Z) белгиси билан белгиланади.

Матн файлга бошқача таъриф бериш ҳам мумкин. Агар файлни матн таҳририда экранга чиқариш ва ўқиш мумкин бўлса, бу матн файл. Клавиатура ҳам компьютерга фақат матнларни жўнатади. Бошқача айтганда программа томонидан экранга чиқариладиган барча маълумотларни *stdout* номидаги матн файлига чиқарилмоқда деб қараш мумкин. Худди шундай клавиатурадан ўқиладиган ҳар қандай берилганларни матн файлидан ўқилмоқда деб ҳисобланади.

Матн файлларининг компоненталари *сатрлар* деб номланади. Сатрлар узлуксиз жойлашиб, турли узунликда ва бўш бўлиши мумкин. Фараз қилайлик, *T* матн файли 4 сатрдан иборат бўлсин:

1- satr#13#10	2- satr uzunroq #13#10	#13#10	4-satr#13#10#26
---------------	------------------------	--------	-----------------

11.3-расм. Тўртта сатрдан ташкил топган матн файли

Матнни экранга чиқаришда сатр охиридаги #13#10 бошқарув белгилари жуфтлиги курсорни кейинги қаторга туширади ва уни сатр бошига олиб келади. Бу матн файл экранга чоп этилса, унинг кўриниши куйидагича бўлади:

```
1- satr[13][10]
2- satr uzunroq[13][10]
[13][10]
4- satr[13][10]
[26]
```

Бу ерда *[n]* - *n*- кодли бошқарув белгисини билдиради. Одатда матн таҳрирлари бу белгиларни кўрсатмайди.

*Бинар файллар* - бу оддийгина байтлар кетма-кетлиги. Одатда бинар файллардан берилганларни фойдаланувчи томонидан бевосита «кўриш»

зарур бўлмаган ҳолларда ишлатилади. Бинар файллардан ўқиш-ёзишда байтлар устида ҳеч қандай конвертация амаллари бажа-рилмайди.

**38-мавзу:** Стандарт оқимлар. Оқимлар билан ишлаш. Ўқиш - ёзиш функциялари.

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

- 1.С++ тилидаги программа тузилиши ва унинг компиляцияси
- 2.С++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

### **Ўқиш-ёзиш оқимлари. Стандарт оқимлар**

Оқим тушунчаси берилганларни файлга ўқиш-ёзишда уларни белгилар кетма-кетлиги ёки оқими кўринишида тасаввур қилишдан келиб чиққан. Оқим устида қуйидаги амалларни бажариш мумкин:

- оқимдан берилганлар блокини оператив хотирага ўқиш;
- оператив хотирадаги берилганлар блокини оқимга чиқариш;
- оқимдаги берилганлар блокини янгилаш;
- оқимдан ёзувни ўқиш;
- оқимга ёзувни чиқариш.

Оқим билан ишлайдиган барча функциялар буферли, формат-лашган ёки форматлашмаган ўқиш-ёзишни таъминлайди.

Программа ишга тушганда ўқиш-ёзишнинг қуйидаги стандарт оқимлар очилади:

*stdin* - ўқишнинг стандарт воситаси;  
*stdout* - ёзишнинг стандарт воситаси;  
*stderr* - хатолик ҳақида хабар беришнинг стандарт воситаси;  
*stdprn* - қоғозга чоп қилишнинг стандарт воситаси;  
*stdaux* - стандарт ёрдамчи қурилма.

Келишув бўйича *stdin* - фойдаланувчи клавиатураси, *stdout* ва *stderr*- терминал (экран), *stdprn*- принтер билан, ҳамда *stdaux*- компьютер ёрдамчи портларига боғланган ҳисобланади. Берилган-ларни ўқиш-ёзишда *stderr* ва *stdaux* оқимидан бошқа оқимлар буфер-ланади, яъни белгилар кетма-кетлиги оператив хотиранинг буфер деб номланувчи соҳасида вақтинча жамланади. Масалан, белгиларни ташқи қурилмага чиқаришда белгилар кетма-кетлиги буферда жамланади ва буфер тўлгандан кейингина ташқи қурилмага чиқарилади.

Ҳозирдаги операцион системаларда клавиатура ва дисплейлар матн файллари сифатида қаралади. Ҳақиқатдан ҳам берилганларни

клавиатурадан программага киритиш (ўқиш) мумкин, экранга эса чиқариш (ёзиш) мумкин. Программа ишга тушганда стандарт ўқиш ва ёзиш оқимлари ўрнига матн файлларни тайинлаш орқали бу оқим-ларни қайта аниқлаш мумкин. Бу ҳолатга *ўқишни (ёзишни) қайта адреслаш рўй берди* дейилади. Ўқиш учун қайта адреслашда ‘<’ белгисидан, ёзиш учун эса ‘>’ белгисидан фойдаланилади. Мисол учун “gauss.exe” бажарилувчи программа берилганларни ўқишни клавиатурадан эмас, балки “massiv.txt” файлидан амалга ошириш зарур бўлса, у буйруқ сатрида қуйидаги кўринишда юкланиши зарур бўлади:

```
gauss.exe < massiv.txt
```

Агар программа натижасини “natija.txt” файлига чиқариш зарур бўлса

```
gauss.exe > natija.txt
```

сатри ёзилади.

Ва ниҳоят, агар берилганларни “massiv.txt” файлидан ўқиш ва натижани “natija.txt” файлига ёзиш учун

```
gauss.exe < massiv.txt > natija.txt
```

буйруқ сатри терилади.

Умуман олганда, бир программанинг чиқиш оқимини иккинчи программанинг кириш оқими билан боғлаш мумкин. Буни *конвейрли жўнатиш* дейилади. Агар иккита junat.exe программаси qabul.exe программасига берилганларни жўнатиши керак бўлса, у ҳолда улар ўртасига ‘|’ белги қўйиб ёзилади:

```
junat.exe | qabul.exe
```

Бу кўринишдаги программалар ўртасидаги конвейрли жўна-тишни операцион системанинг ўзи таъминлайди.

### **Белгиларни ўқиш-ёзиш функциялари**

Белгиларни ўқиш-ёзиш функциялари макрос кўринишида амалга оширилган.

*getc()* макроси тайинланган оқимдан навбатдаги белгини қайта-ради ва кириш оқими кўрсаткичини кейинги белгини ўқишга мосла-ган ҳолда оширади. Агар ўқиш муваффақиятли бўлса *getc()* функция-си ишорасиз *int* кўринишидаги қийматни, акс ҳолда *EOF* қайтаради. Ушбу функция прототипи қуйидагича:

```
int getc(FILE * stream)
```

*EOF* идентификатор макроси

```
#define EOF(-1)
```

кўринишида аниқланган ва ўқиш-ёзиш амалларида файл охирини белгилаш учун хизмат қилади. *EOF* қиймати ишорали *char* турида деб

ҳисобланади. Шу сабабли ўқиш-ёзиш жараёнида *unsigned char* туридаги белгилар ишлатилса, *EOF* макросини ишлатиб бўлмайди.

Навбатдаги мисол *getc()* макросини ишлатишни намоён қилади.

```
#include <iostream.h>
#include <stdio.h>
int main()
{
    char ch;
    cout<<"Belgini kiriting: ";
    ch=getc(stdin);
    cout<<"Siz "<<ch<<" belgisini kiritdingiz.\n";
    return 0;
}
```

*getc()* макроси аксарият ҳолатларда *stdin* оқими билан ишлатилганлиги сабабли, унинг *getc(stdin)* кўринишига эквивалент бўлган *int getchar()* макроси аниқланган. Юқоридаги мисолда “*ch=getc(stdin);*” қаторини “*ch=getchar();*” қатори билан алмаштириш мумкин.

Белгини оқимга чиқариш учун *putc()* макроси аниқланган ва унинг прототипи

```
int putc(int c,FILE*stream)
```

кўринишида аниқланган. *putc()* функцияси *stream* номи билан берилган оқимга *c* белгини чиқаради. Функция қайтарувчи қиймати сифатида *int* турига айлантирилган *c* белги бўлади. Агар белгини чиқаришда хатолик рўй берса *EOF* қайтарилади.

*putc()* функциясини стандарт *stdout* оқими билан боғланган ҳолати - *putc(c,stdout)* учун *putchar(c)* макроси аниқланган.

### 39-мавзу: Бинар файллар

Ажратилган соат:	2 соат
Машғулот тури:	маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

- 1.С++ тилидаги программа тузилиши ва унинг компиляцияси
- 2.С++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

*Бинар файллар* - бу оддийгина байтлар кетма-кетлиги. Одатда бинар файллардан берилганларни фойдаланувчи томонидан бевосита «кўриш» зарур бўлмаган ҳолларда ишлатилади. Бинар файллардан

ўқиш-ёзишда байтлар устида ҳеч қандай конвертация амаллари бажарилмайди.

#### **40-мавзу: Форматли ўқиш ва ёзиш функциялари**

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

- 1.C++ тилидаги программа тузилиши ва унинг компиляцияси
- 2.C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

#### **Форматли ўқиш ва ёзиш функциялари**

Форматли ўқиш ва ёзиш функциялари - *scanf()* ва *printf()* функциялари C тилидан ворислик билан олинган. Бу функцияларни ишлатиш учун “stdio.h” сарлавҳа файлини программага қўшиш керак бўлади.

Форматли ўқиш функцияси *scanf()* қуйидаги прототипга эга:

```
int scanf(const char * <формат>[<адрес>,...])
```

Бу функция стандарт оқимдан берилганларни форматли ўқишни амалга оширади. Функция, кириш оқимидаги майдонлар кетма-кет-лиги кўринишидаги белгиларни бирма-бир ўқийди ва ҳар бир майдонни <формат> сатрида келтирилган формат аниқлаштирувчи-сига мос равишда форматлайди. Оқимдаги ҳар бир майдонга формат аниқлаштирувчиси ва натижа жойлашадиган ўзгарувчининг адреси бўлиши шарт. Бошқача айтганда, оқимдаги майдон (ажратилган белгилар кетма-кетлиги) кўрсатилган форматдаги қийматга аксланти-рилади ва ўзгарувчи билан номланган хотира бўлагига жойлашти-рилади (сақланади). Функция оқимдан берилганларни ўқиш жараё-нини «тўлдирувчи белгини» учратганда ёки оқим тугаши натижасида тўхтатиши мумкин. Оқимдан берилганларни ўқиш муваффақиятли бўлса, функция муваффақиятли айлантирилган ва хотирага сақланган майдонлар сонини қайтаради. Агар ҳеч бир майдонни сақлаш имкони бўлмаган бўлса, функция 0 қийматини қайтаради. Оқим охирига келиб қолганда (файл ёки сатр охирига) ўқишга ҳаракат бўлса, функция *EOF* қийматини қайтаради.

Форматлаш сатри - <формат> белгилар сатри бўлиб, у учта тоифага бўлинади:

- тўлдирувчи белгилар;



- тўлдирувчи белгилардан фарқли белгилар;
- формат аниқлаштирувчилари.

*Тўлдирувчи-белгилар* – бу пробел, ‘\t’, ‘\n’ белгилари. Бу белгилар форматлаш сатридан ўқилади, лекин сақланмайди.

*Тўлдирувчи белгилардан фарқли белгилар* – бу қолган барча ASCII белгилари, ‘%’ белгисилан ташқари. Бу белгилар форматлаш сатридан ўқилади, лекин сақланмайди.

11.1–жадвал. Формат аниқлаштирувчилари ва уларнинг вазифаси

Компонента	Бўлиши шарт ёки йўқ	Вазифаси
[*]	Йўқ	Навбатдаги кўриб чиқиладиган майдон қийматини ўзгарувчига ўзлаштирмаслик белгиси. Кириш оқимидаги майдон кўриб чиқилади, лекин ўзгарувчида сақланмайди.
[<кенглик>]	Йўқ	Майдон кенглигини аниқлаштирувчиси. Ўқиладиган белгиларнинг максимал сонини аниқлайди. Агар оқимда тўлдирувчи белги ёки алмаштирилмайдиган белги учраси функция нисбатан кам сондаги белгиларни ўқиши мумкин.
[F N]	Йўқ	Ўзгарувчи кўрсаткичининг (адресининг) модификатори: F– far pointer; N–near pointer
[h l L]	Йўқ	Аргумент турининг модификатори. <тур белгиси> билан аниқланган ўзгарувчининг қисқа (short - h) ёки узун (long –l,L) кўри–нишини аниқлайди.
<тур белгиси>	Ҳа	Оқимдаги белгиларни алмаштириладиган тур белгиси

*Формат аниқлаштирувчилари* – оқим майдонидаги белгиларни кўриб чиқиш, ўқиш ва адреси билан берилган ўзгарувчилар турига мос равишда алмаштириш жараёнини бошқаради. Ҳар бир формат аниқлаштирувчисига битта ўзгарувчи адреси мос келиши керак. Агар формат аниқлаштирувчилари сони ўзгарувчилардан кўп бўлса, натижа нима бўлишини олдиндан айтиб бўлмайди. Акс ҳолда, яъни ўзгарувчилар сони кўп бўлса, ортиқча ўзгарувчилар инobatга олинмайди.

Формат аниқлаштирувчиси қуйидаги кўринишга эга:

% [\*][<кенглик>] [F|N] [h|l|L] <тур белгиси>

Формат аниқлаштирувчиси ‘%’ белгисидан бошланади ва ундан кейин 11.1–жадвалда келтирилган шарт ёки шарт бўлмаган компонентлар келади.

11.2–жадвал. Алмаштириладиган тур аломати белгилари

Тур аломати	Кутилаётган қиймат	Аргумент тури
<b>Сон туридаги аргумент</b>		
<i>d, D</i>	Ўнлик бутун	<i>int * arg</i> ёки <i>long * arg</i>
<i>E, e</i>	Сузувчи нуқтали сон	<i>float * arg</i>
<i>F</i>	Сузувчи нуқтали сон	<i>float * arg</i>
<i>G, g</i>	Сузувчи нуқтали сон	<i>float * arg</i>
<i>O</i>	Саккизлик сон	<i>int * arg</i>
<i>O</i>	Саккизлик сон	<i>long * arg</i>
<i>I</i>	Ўнлик, саккизлик ва ўн олтилик бутун сон	<i>int * arg</i>
<i>I</i>	Ўнлик, саккизлик ва ўн олтилик бутун сон	<i>long * arg</i>
<i>U</i>	Ишорасиз ўнлик сон	<i>unsigned int * arg</i>
<i>U</i>	Ишорасиз ўнлик сон	<i>unsigned long * arg</i>
<i>X</i>	Ўн олтилик сон	<i>int * arg</i>
<i>X</i>	Ўн олтилик сон	<i>int * arg</i>
<b>Белгилар</b>		
<i>S</i>	Сатр	<i>char * arg</i> (белгилар массиви)
<i>C</i>	Белги	<i>char * arg</i> (белги учун майдон кенглиги берилиши мумкин (масалан, %4с). <i>N</i> белгидан ташкил топган белгилар массивига кўрсаткич: <i>char arg[N]</i> )
<i>%</i>	‘%’ белгиси	Ҳеч қандай алмаштиришлар бажарилмайди, ‘%’ белгиси сақланади.
<b>Кўрсаткичлар</b>		
<i>N</i>	<i>int * arg</i>	<i>%n</i> аргументигача муваффақиятли ўқилган белгилар сони, айнан шу <i>int</i> кўрсаткичи бўйича адресда сақланади.
<i>P</i>	<i>YYYY:ZZZZ</i> ёки <i>ZZZZ</i> кўринишидаги ўн олтилик	Объектга кўрсаткич ( <i>far*</i> ёки <i>near*</i> ).

Оқимдаги белгилар бўлагини алмаштириладиган тур аломати-нинг қабул қилиши мумкин бўлган белгилар 12.2-жадвалда келтирилган.

11.3–жадвал. Формат аниқлаштирувчилари ва уларнинг вазифаси

Компонента	Бўлиши шарт ёки йўқ	Вазифаси
[байроқ]	Йўқ	Байроқ белгилари. Чиқарилаётган қийматни чапга ёки ўнга текислашни, соннинг ишорасини, ўнлик каср нуқтасини, охирдаги нолларни, саккизлик ва ўн олтилик сонларнинг аломатларни чоп

		этишни бошқаради. Масалан, ‘-‘ байроғи қийматни ажратилган ўринга нисбатан чапдан бошлаб чиқаришни ва керак бўлса ўнгдан пробел билан тўлдиришни билдиради, акс ҳолда чап томондан пробеллар билан тўлдиради ва давомига қиймат чиқарилади.
[<кенглик>]	Йўқ	Майдон кенглигини аниқлаштирувчиси. Чиқариладиган белгиларнинг минимал сонини аниқлайди. Зарур бўлса қиймат ёзилишидан ортган жойлар пробел билан тўлдирилади.
[.<хона>]	Йўқ	Аниқлик. Чиқариладиган белгиларнинг максимал сонини кўрсатади. Сондаги рақамларнинг минимал сонини.
[F N h l L]	Йўқ	Ўлчам модификатори. Аргументнинг қисқа ( <i>short</i> - <i>h</i> ) ёки узун ( <i>long</i> - <i>l,L</i> ) кўринишини, адрес турини аниқлайди.
<тур белгиси>	Ҳа	Аргумент қиймати алмаштириладиган тур аломати белгиси

Форматли ёзиш функцияси *printf()* қуйидаги прототипга эга:

```
int printf(const char * <формат>[,<аргумент>,...])
```

Бу функция стандарт оқимга форматлашган чиқаришни амалга оширади. Функция аргументлар кетма-кетлигидаги ҳар бир аргумент қийматини қабул қилади ва унга <формат> сатридаги мос формат аниқлаштирувчисини қўллайди ва оқимга чиқаради.

11.4–жадвал. *printf()* функциясининг алмаштириладиган тур белгилари

Тур аломати	Кутилаётган қиймат	Чиқиш формати
Сон қийматлари		
<i>D</i>	Бутун сон	Ишорали ўнлик бутун сон
<i>I</i>	Бутун сон	Ишорали ўнлик бутун сон
<i>O</i>	Бутун сон	Ишорасиз саккизлик бутун сон
<i>U</i>	Бутун сон	Ишорасиз ўнлик бутун сон
<i>X</i>	Бутун сон	Ишорасиз ўн олтилик бутун сон ( <i>a,b,c,d,e,f</i> белгилари ишлатилади)
<i>X</i>	Бутун сон	Ишорасиз ўн олтилик бутун сон ( <i>A,B,C,D,E,F</i> белгилари ишлатилади)
<i>F</i>	Сузувчи нуқтали сон	[ <i>-</i> ] <i>dddd.dddd</i> кўринишидаги сузувчи нуқтали сон
<i>E</i>	Сузувчи нуқтали сон	[ <i>-</i> ] <i>d.dddd</i> ёки <i>e[+/-]ddd</i> кўринишидаги сузувчи нуқтали сон

<i>G</i>	Сузувчи нуқтали сон	Кўрсатилган аниқликка мос <i>e</i> ёки <i>f</i> шаклидаги сузувчи нуқтали сон
<i>E, G</i>	Сузувчи нуқтали сон	Кўрсатилган аниқликка мос <i>e</i> ёки <i>f</i> шаклидаги сузувчи нуқтали сон. <i>e</i> формат учун ‘ <i>E</i> ’ чоп этилади.
Белгилар		
<i>S</i>	Сатрга кўрсаткич	0-белгиси учрамагунча ёки кўрсатилган аниқликка эришилмагунча белгилар оқимга чиқарилади.
<i>C</i>	Белги	Битта белги чиқарилади
<i>%</i>	Ҳеч нима	‘ <i>%</i> ’ белгиси оқимга чиқарилади.
Кўрсаткичлар		
<i>N</i>	<i>int</i> кўрсаткич ( <i>int* arg</i> )	<i>%n</i> аргументигача муваффақиятли чиқарилган белгилар сони, айнан шу <i>int</i> кўрсаткичи бўйича адресда сақланади.
<i>P</i>	Кўрсаткич	Аргументни <i>YYYY:ZZZZ</i> ёки <i>ZZZZ</i> кўринишидаги ўн олтилик сонга айлантириб оқимга чиқаради.

Ҳар бир формат аниқлаштирувчисига битта ўзгарувчи адреси мос келиши керак. Агар формат аниқлаштирувчилари сони ўзгарувчи-лардан кўп бўлса, натижада нима бўлишини олдиндан айтиб бўлмайди. Акс ҳолда, яъни ўзгарувчилар сони кўп бўлса, ортиқча ўзгарувчилар инobatта олинмайди. Агар оқимга чиқариш муваффақиятли бўлса, функция чиқарилган байтлар сонини қайтаради, акс ҳолда *EOF*.

*printf()* функциясининг <формат> сатри аргументларни алмаштириш, форматлаш ва берилганларни оқимга чиқариш жараёнини бошқаради ва у икки турдаги объектлардан ташкил топади:

- оқимга ўзгаришсиз чиқариладиган оддий белгилар;
- аргументлар рўйхатидаги танланадиган аргументга қўлланиладиган формат аниқлаштирувчилари.

Формат аниқлаштирувчиси қуйидаги кўринишга эга:

*%* [<байроқ>][<.кенглик>] [.<хона>][F|N|h|l|L] <тур белгиси>

Формат аниқлаштирувчиси ‘*%*’ белгисидан бошланади ва ундан кейин 11.3–жадвалда келтирилган шарт ёки шарт бўлмаган компоненталар келади.

Алмаштириладиган тур белгисининг қабул қилиши мумкин бўлган белгилар 11.4- жадвалда келтирилган.

Берилганлар қийматларини оқимдан ўқиш ва оқимга чиқаришда *scanf()* ва *printf()* функцияларидан фойдаланишга мисол:

```
#include <stdio.h>
int main()
{
```

```

    int bson, natija;
    float hson;
    char blg, satr[81];
    printf("\nButun va suzuvchi nuqtali sonlarni,");
    printf("\nbelgi hamda satrni kiriting\n");
    natija=scanf("%d      %f      %c      %s",      &bson,
&hson,&blg,satr);
    printf("\nOqimdan      %d      ta      qiymat      o'qildi
",natija);
    printf("va ular quyidagilar:");
    printf("\n %d %f %c %s \n",bson, hson, blg,
satr);
    return 0;
}

```

Программа фойдаланувчидан бутун ва сузувчи нуқтали сонларни, белги ва сатрни киритишни сўрайди. Бунга жавобан фойдаланувчи томонидан

**10 12.35 A Satr**

қийматлари киритилса, экранга

**Oqimdan 4 ta qiymat o'qildi va ular quyidagilar:**  
**10 12.35 A Satr**

сатрлари чоп этилади.

#### **41-мавзу: Динамик тузилмалар**

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

**ДАРС Р Е Ж АСИ (асосий саволлар):**

- 1.С++ тилидаги программа тузилиши ва унинг компиляцияси
- 2.С++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

Берилганлар устида ишлашда уларнинг миқдори қанча бўлиши ва уларга хотирадан қанча жой ажратиш кераклиги олдиндан номаълум бўлиши мумкин. Программа ишлаш пайтида берилганлар учун зарурат бўйича хотирадан жой ажратиш ва уларни кўрсаткичлар билан боғлаш орқали ягона тузилмалар ҳосил қилиш жараёни *хотиранинг динамик тақсими* дейилади. Бу усулда ҳосил бўлган берилганлар мажмуасига *берилганларнинг динамик тузилмалари* дейилади, чунки уларнинг ўлчами

программа бажарилишида ўзгариб туради. Программалашда динамик тузилмалардан чизиқли рўйхатлар (занжирлар), стеклар, навбатлар ва бинар дарахтлар нисбатан кўп ишлатилади. Улар бир - биридан элементлар ўртасидаги боғланиш-лари ва улар устида бажариладиган амаллари билан фарқланади. Программа ишлашида тузилмаларга янги элементлар қўшилиши ёки мавжудлари ўчирилиши мумкин.

Ҳар қандай берилганларнинг динамик тузилмаси майдонлардан ташкил топади ва уларнинг айримлари қўшни элементлар билан боғланиш учун хизмат қилади.

**Масала.** Нолдан фарқли бутун сонлардан иборат чизиқли рўйхат яратилсин ва ундан кўрсатилган сонга тенг элемент ўчирил-син.

Бутун сонларнинг чизиқли рўйхат кўринишидаги динамик тузилмалари қуйидаги майдонлардан ташкил топади:

```
struct Zanjir
{
    int element;
    Zanjir * keyingi;
};
```

Программа матни:

```
#include <iostream.h>
struct Zanjir
{
    int element;
    Zanjir*keyingi;
};
Zanjir      *      Element_Joylash(Zanjir*z,      int
yangi_elem)
{
    Zanjir*yangi=new Zanjir;
    yangi->element=yangi_elem;
    yangi->keyingi=0;
    if(z)          // рўйхат бўш эмас
    {
        Zanjir*temp=z;
        while(temp->keyingi)
            temp=temp->keyingi;//      рўйхатнинг      охириги
элементини
                                // топиш
        temp->keyingi=yangi;// янги элементни рўйхат
                                // охирига қўшиш
    }
    else z=yangi;    // рўйхат бўш
    return z;        // рўйхат боши адресини қайтариш
}
```

```

Zanjir * Element_Uchirish(Zanjir*z, int del_elem)
{
    if(z)
    {
        Zanjir*temp=z;
        Zanjir*oldingi=0;    // жорий элементдан олдинги
                             // элементга кўрсаткич
        while(temp)
        {
            if(temp->element==del_elem)
            {
                if(oldingi) //ўчириладиган элемент биринчи
эмас
                {
                    // ўчириладиган элементдан олдинги элементни
                    // кейинги элементга улаш
                    oldingi->keyingi=temp->keyingi;
                    delete temp;    // элементни ўчириш
                    temp=oldingi->keyingi;
                }
            }
            else
            {
                // ўчириладиган элемент рўйхат бошида
                z=z->keyingi;
                delete temp;
                temp=z;
            }
        }
        else // элемент ўчириладиган сонга тенг эмас
        {
            oldingi=temp;
            temp=temp->keyingi;
        }
    }
    return z;
}

void Zanjir_Ekranga(Zanjir*z)
{
    cout<<"Zanjir elementlari:"<<endl;
    Zanjir*temp=z;
    while(temp)
    {
        cout<<temp->element<<' ';
        temp=temp->keyingi;
    }
}

```

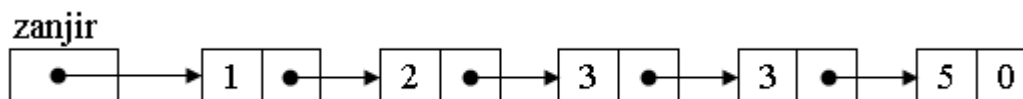
```

    }
    cout<<endl;
}
Zanjir*Zanjirni_Uchirish(Zanjir*z)
{
    Zanjir*temp=z;
    while(z)
    {
        z=z->keyingi;
        delete temp;
        temp=z;
    }
    return z;
}
int main()
{
    Zanjir*zanjir=0;
    int son,del_element;
    do
    {
        cout<<"\nSonni kiriting (0-jaryon tugatish): ";
        cin>>son;
        if(son) zanjir=Element_Joylash(zanjir,son);
    }
    while(son);
    Zanjir_Ekranga(zanjir);
    cout<<"\nO'chiriladigan elementni kiriting: ";
    cin>>del_element;
    zanjir=Element_Uchirish(zanjir,del_element);
    Zanjir_Ekranga(zanjir);
    Zanjir=Zanjirni_Uchirish(zanjir);
    return 0;
}

```

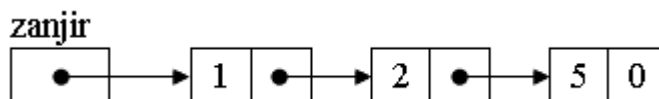
Программанинг бош функциясида чизикли рўйхат ҳосил қилиш учун *Zanjir* туридаги *zanjir* ўзгарувчиси аниқланган бўлиб, унга бўш кўрсаткич қиймати 0 берилган (унинг эквиваленти - *NULL*). Кейин такрорлаш оператори танасида клавиатурадан бутун сон ўқилади ва *Element\_Joylash()* функциясини чақириш орқали бу сон рўйхатга охирига қўшилади. Функция янги ҳосил бўлган рўйхат бошининг адресини яна *zanjir* ўзгарувчисига қайтаради. Агар клавиатурадан 0 сони киритилса рўйхатни ҳосил қилиш жараёни тугайди. Фараз қилайлик қуйидаги сонлар кетма-кетлиги киритилган бўлсин: 1,2,3,3,5,0. У ҳолда ҳосил бўлган рўйхат қуйидаги кўринишда бўлади (10.1-расм):





10.1-расм. Бешта сондан ташкил топган чизикли рўйхат

Ҳосил бўлган рўйхатни кўриш учун *Zanjir\_Ekranga()* функцияси чақирилади ва экранда рўйхат элементлари чоп этилади. Рўйхат устида амал сифатида берилган сон билан устма-уст тушадиган элементларни ўчириш масаласи қаралган. Бунинг учун ўчириладиган сон *del\_element* ўзгарувчига ўқилади ва у *Element\_Uchirish()* функция-си чақирилишида аргумент сифатида узатилади. Функция бу сон билан устма-уст тушадиган рўйхат элементларини ўчиради (агар бундай элемент мавжуд бўлса) ва ўзгарган рўйхат бошининг адреси-ни *zanjir* ўзгарувчисига қайтариб беради. Масалан, рўйхатдан 3 сони билан устма-уст тушадиган элементлар ўчирилгандан кейин у қуйидаги кўринишга эга бўлади (10.2-расм):



10.2-расм. Рўйхатдан 3 сонини ўчирилгандан кейинги кўриниш

Ўзгарган рўйхат элементлари экранга чоп этилади. Программа охирида, *Zanjirni\_Uchirish()* функциясини чақириш орқали рўйхат учун динамик равишда ажратилган хотира бўшатилади (гарчи бу ишнинг программа тугаши пайтида бажарилишининг маъноси йўқ).

Динамик тузилмалардаги ўзгартиришлар (рўйхатга элемент қўшиш ёки ўчириш) мос статик тузилмаларга нисбатан кам амалларда бажарилиши, улар воситасида масалаларни самарали ечишнинг асосларидан бири ҳисобланади.

#### 42-мавзу: Берилганларнинг динамик тузилмалари: чизикли рўйхатлар, стеклар

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

- 1.С++ тилидаги программа тузилиши ва унинг компиляцияси
- 2.С++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

### **43-мавзу: Навбатлар ва бинар дарахтлар**

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

**ДАРС Р Е Ж АСИ (асосий саволлар):**

1. C++ тилидаги программа тузилиши ва унинг компиляцияси
2. C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

### **44-мавзу: C++ тилида синфлар. Синфни ва объектларни тавсифлаш.**

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

**ДАРС Р Е Ж АСИ (асосий саволлар):**

1. C++ тилидаги программа тузилиши ва унинг компиляцияси
2. C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компаноовка, препроцессор, оқим, компилятор.

#### **Таянч синфлар.**

Таянч синфлар C++ тили ички имкониятларини кенгайтириб дастурчи олдида амалиётнинг мураккаб масалаларини ечишда анча қулайлик яратади. Ушбу мавзуда сиз қуйидагиларни билиб оласиз.

- Синфлар ва объектлар ўзида нимани ифодалайди ?
- Янги синфни ва бу синф объектини қандай ҳосил қилиш керак ?
- Функция аъзолар ва ўзгарувчи аъзолар нима ?
- Конструктор нима ва уни қандай ишлатиш керак ?

#### **Янги тип тузиш.**

Олдинги дарсларда бутун, ҳақиқий ва белгили типлар билан танишган эдик. Улардан маълумки, тип орқали ўзгарувчи хусусиятлари характерланади. Масалан, агар Height ва Width ўзгарувчиларни ишорасиз қисқа бутун (unsigned short int) типда эълон қилсак, уларнинг ҳар бири 0 - 65535 диапазондаги сонларни қабул қилиши мумкин ва бунда улар 2 байтдан жой эгаллайди. Агарда сиз бу

ўзгарувчиларга шу ораликдан ташқари бирор сон бермоқчи бўлсангиз хатолик ҳақида ахборот оласиз.

Демак, Height ва Width ўзгарувчиларини ишорасиз бутун сон деб эълон қилишдан сиз бу ўзгарувчиларни қўшиш ёки уларнинг бирининг қийматини иккинчисига ўзлаштириш имконига эга бўласиз.

Демак ўзгарувчи типи:

- унинг хотирадаги ўлчовини
- у сақлаши мумкин бўлган маълумот типини
- унинг ёрдамида бажариш мумкин бўлган операцияларни аниқлайди.

Берилганларни типи категориялари сифатида автомобил, уй, одам, геометрик фигураларни мисол қилиб келтириш мумкин. C++ тилида дастурчи ўзига керакли ихтиёрий типни ҳосил қилиши мумкин. Бу тип эса ички таянч типларни хоссалари ва функционал имкониятларини ўзида ифодалайди.

### **Нима учун янги тип тузиш керак.**

Одатда дастурлар ходимлар ҳақидаги маълумотларни қайта ишлаш ёки иситиш системаси ишини имитация қилиш каби амалдаги масалаларни ечиш учун ёзилади. Бу масалани фақатгина бутун ёки белгили қийматлар ёрдами билан ҳам ечиш мумкин. Агарда сиз турли объектлар учун улкан типлар ҳосил қилсангиз бу масалаларни ечиш етарлича содда кўринади. Бошқа сўз билан айтганда иситиш системаси ишини имитация қилишда, агарда иссиқлик ўлчагичлар, термостатлар ва бойлерларни ифодаловчи ўзгарувчилар тузилса уни жорий қилиш осонроқ бўлади. Бу ўзгарувчилар реалликка қанчалик яқин бўлса, унинг дастурини тузиш шунчалик осон бўлади.

### **Синфлар ва синф аъзолари.**

Янги тип синфни эълон қилиш билан тузилади. Синф - бу бир – бири билан функционал орқали боғланган ўзгарувчилар ва методлар тўпламидир. Синфларга амалиётдан кўпгина мисоллар келтириш мумкин. Масалан, автомобилни ғилдирак, эшик, ўриндик, ойна ва бошқа қисмлардан ташкил топган коллекция ёки ҳайдаш тезлигини ошириш, тўхтатиш, буриш имкониятларига эга бўлган объект деб тасаввур қилиш мумкин. Автомобил ўзида турли эҳтиёт қисмларни ва уларни функцияларини инкапсуляция қилади. Автомобил каби синфда ҳам инкапсуляция қатор имкониятларни беради. Барча маълумотлар битта объектда йиғилган ва уларга осонгина мурожаат қилиш, уларни ўзгартириш ва кўчириш мумкин. Сизнинг синфингиз билан ишловчи дастурий қисмлар, яъни мижозлар сизнинг объектингиздан, унинг қандай ишлашидан ташвишланмасдан, бемалол фойдаланишлари мумкин.

Синф ўзгарувчиларнинг ихтиёрий комбинациясидан, шунингдек бошқа синфлар типларидан иборат бўлиши мумкин. Синфдаги ўзгарувчилар ўзгарувчи – аъзолар ёки хоссалар дейилади. Car синфи ўриндиқ, радиоприёмник, шина ва бошқа ўзгарувчи - аъзолардан иборат. Ўзгарувчи – аъзолар фақатгина ўзларининг синфларида ётадилар. Ғилдирак ва мотор автомобилнинг қандай таркибий қисми бўлса, ўзгарувчи – аъзолар ҳам синфнинг шундай таркибий қисмидир.

Синфдаги функциялар одатда ўзгарувчи аъзолар устида бирор бир амал бажарадилар. Улар функция – аъзолар ёки синф методлари деб айтилади. Mashina синфи методлари қаторига Haydash() ва Tuxtatish() методлари киради. Mushuk синфи ҳайвонни ёши ва оғирлигини ифодаловчи ўзгарувчи – аъзоларга эга бўлиши мумкин. Шунингдек, бу синфнинг функционал қисми Uxlash(), Miyovlash(), SichqonTutish() методларидан иборат бўлади.

Функция – аъзолар ҳам ўзгарувчи аъзолар сингари синфда ётади. Улар ўзгарувчи аъзолар устида амаллар бажаради ва синфни функционал имкониятларини аниқлайди.

### **Синфни эълон қилиш.**

Синфни эълон қилиш учун class калитли сўзи, ундан сўнг очилувчи фигурали кавс, сўнг хоссалар ва методлари рўйхати ишлатилади. Синфни эълон қилиш ёпилувчи фигурали кавс ва нуқтали вергул орқали яқунланади. Масалан, Mushuk синфини қуйидагича эълон қилиш мумкин.

```
Class Mushuk
{
    unsigned int itsYosh ;
    unsigned int itsOgirlik ;
    void Miyovlash()
}
```

Mushuk синфини эълон қилишда хотира захираланмайди. Эълон қилиш, компиляторга Mushuk синфини мавжудлигини, ҳамда унда қандай қийматлар сақлаши мумкинлиги (itsYosh, itsOgirlik) ва у қандай амалларни бажариши мумкинлиги (Miyovlash() методи) ҳақида хабар беради. Бундан ташқари, бу эълон қилиш орқали компиляторга Mushuk синфининг ўлчами, яъни ҳар бир Mushuk синфи объекти учун компилятор қанча жой ажратиши лозимлиги ҳақида ҳам маълумот беради. Масалан, жорий мисолда бутун қиймат учун тўрт байт талаб қилинса, Mushuk синфи объекти ўлчови саккиз байт бўлади. (itsYosh ўзгарувчиси учун тўрт байт, itsOgirlik ўзгарувчиси учун тўрт байт). Miyovlash() методи хотирадан жой ажратишни талаб қилмайди.

### Объектни эълон қилиш

Янги турдаги объект худди оддий бутун сонли ўзгарувчидек аниқланади. Ҳақиқатан ҳам ихтиёрий бутун сонли ўзгарувчи қуйидагича аниқланади:

```
unsigned int MyVariable  
// ишорасиз бутун сонни аниқлаймиз
```

Cat синфидаги объект эса қуйидагича аниқланади:

```
Mushuk Frisky // Mushuk объектини аниқлаймиз.
```

Бу дастурий кодларда unsigned int типдаги MyVariable номли ўзгарувчи ва Mushuk синфининг Frisky номли объекти аниқланди. Кўпгина ҳолларда синф ва объект тушунчаларини ишлатишда чалкашликка йўл қўйилади. Шунинг учун, объект синфнинг бирор бир экземпляр (нуса) эканлигини яна бир бор таъкидлаш жоиз.

### 45-мавзу: Синф майдонлари ва методлари

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

1. C++ тилидаги программа тузилиши ва унинг компиляцияси
2. C++ тилида бажарилувчи файл яратиш босқичлари

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

### Синф аъзоларига мурожаат қилиш имкони.

Mushuk синфининг реал объектини аниқлаганимиздан сўнг бу объектнинг аъзоларига мурожаат қилиш зарурияти туғилиши мумкин. Бунинг учун бевосита мурожаат ( . ) оператори қўлланилади. Масалан, Frisky объектининг Weight ўзгарувчи - аъзосига 50 сонини ўзлаштирмоқчи бўлсак қуйидаги жумлани ёзишимиз лозим.

```
Fresky.Weight=50;  
Meow () методини чақириш учун эса  
Frisky.Meow ();  
жумласини ёзиш лозим.
```

## Қиймат синфга эмас объектга ўзлаштирилади

C++ тилида берилганлар типига қиймат ўзлаштирилмайди. Қиймат фақатгина ўзгарувчиларга берилади. Масалан, куйидаги ёзув нотўғридир:

```
Int=s // нотўғри
```

Компилятор `int` типига қиймат ўзлаштирилиши хатолик эканлиги ҳақида хабар беради. Худди шу нуктаи – назардан куйидаги ёзув ҳам ноўриндир:

```
Cat.itsYosh= 5 // нотўғри
```

Агарда `Mushuk` объект бўлмасдан синф бўлса, юқоридаги ифодани ҳам компилятор хато деб ҳисоблайди. Ўзлаштириш амалини бажаришда хатоликка йўл қўймаслик учун олдин `Mushuk` синфига тегишли `Frisky` объектини ҳосил қилиш ва унинг `ItsYosh` майдонига 5 қийматини бериш лозим.

```
Mushuk Frisky;  
Frisky.itsYosh=5;
```

## Синф аъзоларига мурожаат қилиш

### имконини чегаралаш.

Синфни эълон қилишда бир нечта калит сўзлардан фойдаланилади. Улардан энг муҳимлари `public` (очиқ) ва `private` (ёпиқ) калит сўзлари бўлиб, улар орқали объектнинг аъзоларига мурожаат қилиш имконияти чегараланади.

Синфнинг барча методлари ва хоссалари бошланғич ҳолда ёпиқ деб эълон қилинади. Ёпиқ аъзоларга фақатгина шу синфнинг методлари орқалигина мурожаат қилиш мумкин. Объектнинг очиқ аъзоларига эса дастурдаги барча функциялар мурожаат қилишлари мумкин. Синф аъзоларига мурожаат қилиш имконини белгилаш жуда муҳим хусусият бўлиб, бу масалани ечишда унча катта тажрибага эга бўлмаган дастурларчилар кўпинча қийинчиликларга дуч келадилар. Бу ҳолатни батафсилроқ тушунтириш учун мавзунини бошида келтирилган масаламизга қайтамыз.

```
Class Mushuk  
{  
    unsigned int itsYosh;  
    unsigned int itsOgirlik;  
    void Miyovlash();  
}
```

Бу тарзда синфни эълон қилишда `itsYosh` ва `itsOgirlik` майдонлари ҳам, `Miyovlash()` методи ҳам ёпиқ аъзо сифатида аниқланади. Дастурда юқоридаги тартибда `Mushuk` синфи эълон қилинган бўлса ва бу синф экземплярлари бўлган объектнинг `itsYosh` аъзосига `main()` функцияси танасидан туриб мурожаат қилсак компилятор хатолик рўй берганлиги ҳақида хабар беради.

```
Mushuk Baroq;
```

```
Baroq.itsYosh = 5 // Хатолик!
// Ёпик аъзога мурожаат қилиш мумкин эмас.
Mushuk синфи аъзоларига дастурнинг бошқа объектлари томонидан
муурожаат қилиш имконини ҳосил қилмоқчи бўлсак, уни public калитли сўзи
орқали амалга оширамиз.
```

```
Class Mushuk
{
    public:
        unsigned int itsYosh;
        unsigned int itsOgirlik;
        void Meow( );
}
```

Энди public калитли сўзи орқали синфнинг барча аъзолари (itsYosh, itsOgirlik, Miyovlash()) очик аъзо бўлди.

6.1 – листингда Mushuk синфи очик ўзгарувчи аъзолари билан эълон қилинган.

### **6.1. – листинг. Оддий синфнинг очик аъзосига мурожаат.**

---

```
1.# include < iostream.h >;
class Mushuk;
{
    public:
        int itsYosh;
        int itsOgirlik;
    }
    int GetYosh(); // Inlizchada Get- olmoq
    void SetYosh (int Age); //Set - o`zgartirmoq
    void Miyovlash();
    private:
        int itsYosh;

int main()
{
    Mushuk Frisky;
    Frisky.itsYosh= 5;
    // ўзгарувчи – аъзога киймат ўзлаштирилди.
    cout << "Frisky " <<Frisky.itsAge;
    cout <<" yoshdagi mushuk.\n";
    return 0;
}
```

---

НАТИЖА:

## ***Frisky 5 yoshdagi mushuk***

### **Синф методларини аниқланиши**

Синф методи аниқлаш учун биринчи синф номи, кейин иккита икки нукта, функция номи ва параметрлари ёзилади. 6.3. – листингда Mushuk синфининг функция аъзосини аниқланиши кўрсатилган.

### **6.2. – листинг. Оддий синфнинг методи аниқланиши.**

---

```
1. #include <iostream.h>
```

```
class Mushuk
{
public:
    int GetYosh();
    void SetYosh(int yosh);
    void Miyovlash();
private:
    int itsYosh;
}

// GetYosh функцияси itsYosh ёпик ўзгарувчи
// аъзонинг кийматини кайтаради
int Mushuk::GetYosh();
{
return itsYosh;
}

// SetYosh функцияси itsYosh ўзгарувчи -
// аъзога киймат ўзлаштиради.
void Mushuk:: SetYosh(int yosh)
{
// itsYosh ўзгарувчи - аъзосига yosh
// киймати параметри ёрдамида берилади.
itsYosh=yosh;
}

void Mushuk::Miyovlash()
{
cout << "Miyov,miyov,miyov.\n";
}

int main( )
```



```

{
    Mushuk Frisky;
    Frisky.SetYosh(5);
    Frisky.Miyovlash();
    cout << "Frisky" <<Frisky.GetYosh();
    cout << " yoshdagi mushuk.\n";
    Frisky.Miyovlash();
    return 0;
}

```

---

НАТИЖА:

```

Miyov, miyov, miyov
Frisky 5 yoshdagi mushuk
Miyov, miyov, miyov

```

#### 46-мавзу: Конструктор ва Деструкторлар

**Ажратилган соат:** 2 соат  
**Машғулот тури:** маъруза

ДАРС Р Е Ж А С И (асосий саволлар):

- 1.Конструкторлар
- 2.Деструкторлар

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

#### Конструкторлар ва деструкторлар

Бутун сонли ўзгарувчини аниқлашнинг икки хил йўли бор. Биринчиси, олдин ўзгарувчини аниқлаш, кейин эса унга бирор бир қиймат ўзлаштиришдир. Масалан,

```

int Ogirlik; // ўзгарувчини аниқлаш
..... // бу ерда бошка ифодалар бор
Ogirlik=7; // ўзгарувчига қиймат ўзлаштирамиз.

```

Иккинчиси, ўзгарувчи аниқланиши билан бирга унга дарҳол қиймат ўзлаштирилади, масалан:

```
int Ogirlik=7;    //ўзгарувчини эълон қиламиз
                //ва унга қиймат ўзлаштирамиз.
```

Қиймат бериш амали ўзгарувчи аниқланиши билан унга бошланғич қиймат ўзлаштирилишини англатади. Кейинчалик, бу ўзлаштирилган қийматни ўзгартиришингиз ҳам мумкин.

Синфнинг ўзгарувчи–аъзосига қандай қиймат ўзлаштирилди? Бунинг учун синфда конструктор деб аталувчи махсус функция – аъзо ишлатилади. Зарурий вақтда конструктор бир нечта параметрни қабул қилади. Лекин ҳеч қандай типдаги қиймат қайтармайди. Конструктор – бу синф номи билан устма – уст тушадиган синф методидир.

Синфда конструкторни эълон қилиниши билан деструкторлар ҳам аниқланиши лозим. Агарда конструктор синф объектини тузиш ва унинг ўзгарувчи – аъзоларига қиймат бериш вазифасини бажарса, деструктор мавжуд объектнинг хотирадан ўчиради. Деструкторлар синф номи олдида тильда (~) белгисини қўйиш орқали аниқланади. Деструкторлар ҳеч қандай аргумент қабул қилмайди ва ҳеч қандай қиймат қайтармайди. Mushuk синфининг деструктори қуйидагича кўринишда аниқланади:

```
~Mushuk()
```

## **Бошланғич берилган конструктор ва деструкторлар**

Агарда сиз конструктор ёки деструкторни аниқламасангиз, сиз учун бу ишни компиляторнинг ўзи бажаради. Стандарт конструктор ва деструкторлар бирорта аргумент қабул қилмайди ва ҳеч қандай амал бажармайди.

### **6.3. – листинг. Конструктор ва деструкторларни қўлланилиши**

```
1. # include <iostream.h>
```

```
class Mushuk
{
public:
    Mushuk(int BoshlYosh);
    ~Cat( );
    int GetYosh();
    void SetYosh(int yosh);
    void Miyovlash();
private:
    int itsYosh;
};

// Mushuk синфи конструктори
Mushuk::Mushuk(int BoshlYosh)
```

```

{
    itsYosh= BoshlYosh;
}

Mushuk::~Mushuk( )
{

}

// GetYosh функцияси itsYosh ўзгарувчи
// аъзосининг кийматини кайтаради
int Mushuk:: GetYosh()
{
    return itsYosh;
}

//SetYosh функцияси itsYosh ўзгарувчи -
// аъзосига киймат ўзлаштиради.
void Mushuk::SetYosh(int yosh)
{
    itsYosh= yosh;
}

// Meow функцияси экранга Miyov ёзувини
// чиқариш учун ишлатилади.
void Mushuk:: Meow()
{
    cout << "Miyov.\n";
}


int main()
{
    Mushuk Frisky(5);
    Frisky.Meow();
    cout << "Frisky"<<Frisky.GetYosh();
    cout <<"yoshdagi mushuk.\n";
    Frisky.Meow();
    Frisky.SetYosh(7);
    cout<< "Hozir Frisky " <<Frisky.GetYosh();
    cout << "yoshda.\n";
    return 0;
}

```

Miyov.  
Frisky 5 yoshda.  
Miyov.  
Hozir Frisky 7 yoshda.

**47-мавзу:** Операторларни қайта юклаш. Ворислик.

**Ажратилган соат:** 2 соат  
**Машғулоти тури:** маъруза

ДАРС Р Е Ж АСИ (асосий саволлар):

1. Операторларни қайта юклаш
2. Ворислик тушунчаси

Мавзудаги асосий таянч тушунчалар ва иборалар: Программа, компиляция, компоновка, препроцессор, оқим, компилятор.

## Amaliy mashg'ulotlar.

### 1. Berilganlar turlari

Ushbu amamliy mashg'ulotlarda berilganlar turiga mos ravishda kompyuter hotirasida egallagan joyi (baytlarda), ularning qiymat diapozonlari va berilganlarni bir turdan ikkinchisiga o'tkazish bilan bog'liq masalalar qaraladi.

Jadvalning har bir satridagi "*Berilgan turi*" ustunidagi turda o'zgaruvchilar e'lon qilinib, ularga mos ravishda 8, 10, va 16 lik sanoq sistemasi ko'rinishidagi uchta qiymat klaviaturadan kiritilsin. Bu qiymatlar jadvalning keyingi ustunlarida keltirilgan turlardagi o'zgaruvchilarga o'zlashtirilsin va chop etilsin (agar mos satr va ustun kesishmasida '+' turgan bo'lsa). Chop etilgan natijalarga izoh berilsin.

#### *Namunaviy masala*

short int turida berilgan o'zgaruvchining qiymatini unsigned char, unsigned short int, long int turidagi o'zgaruvchiga o'zlashtiring va ularning qiymatlariga izoh bering.

#### *Yechish usuli*

Masalani yechishda C++ tilidagi bir turdan ikkinchi turga oshkor va oshkormas keltirish texnologiyasidan foydalaniladi.

#### *Programma matni*

```
#include<iostream.h>
int main()
{
    short int n;
    cout<<"n qiymatini kiriting =";
    cin>>n;
    unsigned char n1;
    unsigned short int n2;
    long int n3;
    n1=n; cout<<"n1="<<int(n1)<<endl;
    n2=n; cout<<"n2="<<n2<<endl;
    n3=n; cout<<"n3="<<n3<<endl;
    return 0;
}
```

Programma ishga tushganda n ning qiymatini kiritish so'raladi va kiritilgan qiymat boshqa turdagi o'zgaruvchilarga o'zlashtiriladi. Hosil bo'lgan natijalar chop qilinadi.

Masalan, n o'zgaruvchining qiymati sifatida -1 soni kiritilsa, quyidagi natijalar chop etiladi:

```
n1=255
n2=65535
```

$n3 = -1$

### Natijalarga izoh

$n = -1$  sonining xotiradagi ichki ko'rinishi "11111111 11111111" ko'rinishida bo'ladi.

$n1$  o'zgaruvchi xotirada 1 bayt egallaydi va u  $n$  o'zgaruvchining kichik baytidagi qiymatni qabul qiladi ( $n1 = 11111111$ ). Bu son ishorasiz butun son va o'nlik sanoq sistemasida 255 soniga teng.

$n2$  o'zgaruvchi xotirada 2 bayt egallaydi va u  $n$  o'zgaruvchining qiymatini to'liq o'zlashtiradi ( $n2 = 11111111 11111111$ ). Bu son ishorasiz butun son va o'nlik sanoq sistemasida 65535 soniga teng.

$n3$  o'zgaruvchi xotirada 4 bayt egallaydi va " $n3 = n$ ;" ko'rsatmasi bajarilishi natijasida  $n3$  o'zgaruvchida -1 sonining qo'shimcha koddagi ko'rinishi hosil bo'ladi ( $n3 = 11111111 11111111 11111111 11111111$ ).

### Amaliy topshiriqlar

Jadvalning "Berilgan turi" ustunida ko'rsatilgan turdagi qiymatni "O'tkaziladigan turlar" turidagi qiymatga o'tkazilsin va natija chop etilsin.

№	Berilgan turi	O'tkaziladigan turlar											
		unsigned				short int	int	long int	float	double	bool	char	
		short int	int	long int	char								
1	short int	+	+	—	+	—	+	—	+	—	+	+	
2	Int	+	+	+	+	+	—	—	+	+	+	+	
3	long int	+	+	+	+	+	+	—	+	+	—	+	
4	float	+	+	+	+	+	+	+	—	—	+	+	
5	double	+	+	+	+	+	+	+	+	—	+	+	
6	long double	+	+	+	+	+	+	+	+	+	—	+	
7	char	+	+	—	+	+	—	—	+	—	+	—	
8	unsigned short int	—	+	—	+	+	+	—	+	—	+	+	
9	unsigned int	+	—	—	+	+	+	—	+	—	—	+	
10	unsigned long int	+	+	—	+	+	+	+	+	+	—	+	
11	unsigned char	+	+	—	—	+	—	—	+	—	+	+	
12	bool	+	+	—	+	+	+	—	+	—	—	+	

**Izoh.** Jadval kataklaridagi '+' belgisi qiymatni o'tkazish kerakligini, '-' belgisi esa qiymatni o'tkazish shart emasligini bildiradi.

## 2. Razryadlar ustida mantiqiy amallar

Berilgan  $a, b, n$  bayt kattaligidagi ishorali butun sonlar ustida quyidagi amallar bajarilsin:  $a \& b$ ,  $a | b$ ,  $a \wedge b$ ,  $\sim a \& b$ ,  $\sim b | a$ ,  $a = a >> n$ ,  $a = a << n$ .

### Namunaviy masala

Bayt kattaligidagi ishorali butun  $a, b$  sonlar ustida razryadli mantiqiy ko'paytirish amali bajarilsin va natijaga izoh berilsin.

### ***Yechish usuli***

Masalani yechishda C++ tilida aniqlangan mantiqiy razryadli amallarga tayaniladi.

### ***Programma matni***

```
#include <iostream.h>
int main()
{
    char a=124, b=10;
    cout<<"\na&b="<<int(a&b);
    return 0;
}
```

Programma ishlashi natijasida ekranga

8

qiymati chop etiladi.

### ***Natijani izohlash***

Natijani izohlashda razryadli mantiqiy amallar aniqlanish qoidasiga asoslanadi.

### **A va B razryadlar ustida mantiqiy amallar jadvali**

A	B	A&B	A B	A^B	~A
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

Berilgan  $a$  va  $b$  o'zgaruvchilar ( $a=124$ ,  $b=10$ ) qiymatlarining bayt razryadlaridagi ikkilik ko'rinishi mos ravishda 01111100 va 00001010 ko'rinishda bo'ladi.

$a$  :01111100 (124)

$b$  :00001010 (10)

$a \& b$ :00001000 (8)

### ***Amaliy topshiriqlar***

№	1	2	3	4	5	6	7	8	9	10
A	2	5	-10	64	28	55	10	-1	15	123
B	-3	125	10	-1	-16	-55	105	126	127	64
N	2	3	4	3	4	2	5	4	3	5

№	11	12	13	14	15	16	17	18	19	20
A	10	127	123	12	126	125	-12	60	20	32
B	-45	100	-100	-1	2	25	-100	15	-20	-32

N	4	7	5	3	4	3	2	4	2	2
---	---	---	---	---	---	---	---	---	---	---

### 3. Matematik ifodalarni hisoblash

Amaliy topshiriqlarni bajarishda matematik funksiyalar va ifodalarning C++ tilida yozilishi, butun va haqiqiy sonlar ustida arifmetik amallar bajarilishiga e'tibor berish kerak bo'ladi. Masalalar test ko'rinishida beriladi, yani olingan natija oldindan berilgan qiymat bilan mos tushishi kerak bo'ladi.

#### *Namunaviy masala*

Haqiqiy turdagi x, y va z o'zgaruvchilarning x=182.5, y=18.225 va z=-0.3298 qiymatlarida

$$a = \left| x^{\frac{y}{x}} - 3\sqrt{\frac{y}{x}} \right| + (y-x) \frac{\cos y - \frac{z}{(y-x)}}{1 + (y-x)^2}$$

ifoda hisoblansin va natijani 1.2132 qiymat bilan ustma-ust tushishi tekshirilsin.

#### *Yechish usuli*

Amallar prioritetiga (ustunlik darajasiga) rioya qilgan holda matematik ifodalarni C++ tilida yozish qoidalari qo'llaniladi. Murakkab matematik ifodaning har bir ifoda osti alohida hisoblanib, oraliq o'zgaruvchilarga o'zlashtiriladi va ular asosida yakuniy natija olinadi. Bu usulning afzalligi shundaki, oraliq natijalar to'g'ri hisoblanganligini nazorat qilish, xatolarni lokallashtirish va ularni bartaraf etish nisbatan onson kechadi.

Standart matematik funksiyalar tilning <math.h> kutubxonasida joylashgan (2-ilova).



### ***Programma matni***

```
#include<iostream.h>
#include<math.h>
int main()
{
    float x,y,z,a,a1,a2,k,t;
    x=182.5;
    y=18.225;
    z=-03.298;
    k=y/x;
    t=y-x;
    a1=fabs(pow(x,k)-pow(k,1./3));
    a2=(cos(y)-z/t)/(1+pow(t,2));
    a=a1+t*a2;
    cout<<"a="<<a;
    return 0;
}
```

Programmaning bajarilishi natijasida ekranga quyidagi natija chop etiladi:

a=1.2132

### ***Amaliy topshiriqlar***

Quyidagi ifodalar o'zgaruvchilarning berilgan qiymatlari uchun hisob-lansin:

1. Berilganlar:  $x=14.26, y=-1.22,$   
 $z=3.5 \times 10^{-2},$

$$t = \frac{2 \cos\left(x - \frac{\pi}{6}\right)}{0.5 + \sin^2 y} \left(1 + \frac{z^2}{3 - z^2/5}\right)$$

Natija:  $t=0.564849$

3. Berilganlar:  $x=-4.5, y=0.75 \times 10^{-4},$   
 $z=0.845 \times 10^2,$

$$u = \frac{\sqrt[3]{8 + |x-y|^2 + 1}}{x^2 + y^2 + 2} - e^{|x-y|} (\lg^2 z + 1)^x.$$

Natija:  $u=-55.6848$

5. Berilganlar:  $x=-15.246, y=4.642 \times 10^{-2},$   
 $z=20.001 \times 10^2,$

$$\alpha = \ln\left(y^{-\sqrt{|x|}}\right) \left(x - \frac{y}{2}\right) + \sin^2 \arctg(z).$$

Natija:  $\alpha=-182.036$

7. Berilganlar:  $x=0.1722, y=6.33,$   
 $z=3.25 \times 10^{-4},$

2. Berilganlar:  $x=3.74 \times 10^{-2},$   
 $y=-0.825, z=0.16 \times 10^2,$

$$v = \frac{1 + \sin^2(x+y)}{\left|x - \frac{2y}{1+x^2y^2}\right|} x^{|y|} + \cos^2\left(\arctg \frac{1}{z}\right)$$

Natija:  $v=1.0553$

4. Berilganlar:  $x=0.4 \times 10^4, y=-0.875,$   
 $z=-0.475 \times 10^{-3},$

$$w = |\cos x - \cos y|^{(1+2\sin^2 y)} \left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4}\right).$$

Natija:  $w=1.9873$

6. Berilganlar:  $x=16.55 \times 10^{-3}, y=-2.75,$   
 $z=0.15,$

$$\beta = \sqrt{10(\sqrt[3]{x} + x^{y+2})} (\arcsin^2 z - |x-y|).$$

Natija:  $\beta=-38.902$

8. Berilganlar:  $x=-2.235 \times 10^{-2}, y=2.23,$   
 $z=15.221$

$$\gamma = 5 \arctg x \times \frac{1}{4} \arccos \frac{x + 3|x-y| + x^2}{|x-y|z + x^2}.$$

Natija:  $\gamma = -172.025$

9. Berilganlar:  $x=1.825 \times 10^2, y=18.225, z=-3.298 \times 10^{-2}$ ,

$$\psi = \left| x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}} \right| + (y-x) \frac{\cos y - \frac{z}{(y-x)}}{1 + (y-x)^2}.$$

Natija:  $\psi = 1.2131$

11. Berilganlar:  $x=6.251, y=0.827, z=25.001$ ,

$$b = y^{\sqrt[3]{|x|}} + \cos y \frac{|x-y| \left( 1 + \frac{\sin^2 z}{\sqrt{x+y}} \right)}{e^{|x-y|} + x/2}.$$

Natija:  $b=0.7121$

13. Berilganlar:  $x=17.421, y=10.365 \times 10^{-3}, z=0.828 \times 10^5$ ,

$$f = \frac{\sqrt[4]{y + \sqrt[3]{x-1}}}{|x-y|(\sin^2 z + \operatorname{tg} z)}.$$

Natija:  $f=0.33056$

15. Berilganlar:  $x=2.444, y=0.869 \times 10^{-2}, z=-0.13 \times 10^3$ ,

$$h = \frac{x^{y+1} + e^{y-1}}{1 + x|y - \operatorname{tg} z|} (1 + |y-x|) + \frac{|y-x|^2}{2} - \frac{|y-x|^3}{3}.$$

Natija:  $h=-0.49871$

17. Berilganlar:  $x=1, y=1, z=3$

$$a = (1+y) \frac{x+y/(x^2+4)}{e^{-x-2} + 1/(x^2+4)};$$

$$b = \frac{1 + \cos(-2)}{x^4/2 + \sin^2 z}.$$

Natija:  $a=9.608184; b=2.962605$

19. Berilganlar:  $x=3, y=4, z=5$ ,

$$a = \frac{1 + \sin^2(x+y)}{2 + |x + 2x/(1+x^2y^2)|} + x;$$

$$\varphi = \frac{e^{|x-y|} |x-y|^{x+y}}{\arctg x \arctg z} + \sqrt[3]{x^6 + \ln^2 y}.$$

Natija:  $\varphi=39.374$

10. Berilganlar:  $x=3.981 \times 10^{-2}, y=-1.625 \times 10^3, z=0.512$ .

$$a = 2^{-x} \sqrt{x + \sqrt[4]{|y|}} \sqrt[3]{e^{x-1/\sin z}}.$$

Natija:  $a=1.26185$

12. Berilganlar:  $x=3.251, y=0.325, z=0.466 \times 10^{-4}$ ,

$$c = 2^{(y^x)} + (3^x)^y - \frac{y \left( \arctg z \frac{\pi}{6} \right)}{|x| + \frac{1}{y^2 + 1}}.$$

Natija:  $c=4.025$

14. Berilganlar:  $x=12.3 \times 10^{-1}, y=15.4, z=0.252 \times 10^3$ ,

$$g = \frac{y^{x+1}}{\sqrt[3]{|y-2|} + 3} + \frac{x + \frac{y}{2}}{2|x+y|} (x+1)^{-1/\sin z}.$$

Natija:  $g=82.8257$

16. Berilganlar:  $x=-1, y=-1, z=3$

$$a = \left( \sqrt{|x-1|} - \sqrt[3]{|y|} \right) \left( 1 + \frac{x^2}{2} + \frac{y^2}{4} \right);$$

$$b = x(\arctg z) + e^{-(x+3)}.$$

Natija:  $a=0.2366935; b=-1.384381$

18. Berilganlar:  $x=3, y=4, z=5$

$$a = \frac{2 \cos(-\pi/6)}{1/2 + \sin^2 y};$$

$$b = 1 + \frac{z^2}{3 + z^2/5}.$$

Natija:  $a=-1.467187; b=4.125$

20. Berilganlar:  $x=5, y=4.5, z=0.5$ ,

$$a = \frac{\sin(xy - e^x)^2}{1 + \frac{x}{y} 2.05 + 0.001 e^{x^2}};$$

$$b = \cos^2(\arctg \frac{1}{z}).$$

Natija: a=3.288716; b=0.9615385

$$b = \sqrt{x^2 - y^2} \ln z + \frac{\lg(y+z)}{\sqrt{x^2 - y^2 + 1}}.$$

Natija: a=8.46425e-10, b= -1.29084

#### 4. Hisoblashga doir sodda masalalar

##### *Namunaviy masala*

O‘zaro teng bo‘lmagan  $x$  va  $y$  o‘zgaruvchilar berilgan. Bu o‘zgaruvchilarning qiymatlari qo‘shimcha o‘zgaruvchidan foydalanmagan holda almashtirilsin.

##### *Yechish usuli*

$x$  va  $y$  o‘zgaruvchilar ustida qo‘shish va ayirish amallarini ketma-ket bajarish orqali ularning qiymatlari o‘zaro almashtiriladi.

##### *Programma matni*

```
#include<iostream.h>
#include<math.h>
int main()
{
    float x,y,z;
    cout<<"x="; cin>>x; // masalan x=5;
    cout<<"y="; cin>>y; // masalan y=7;
    x+=y; // 12
    y=x-y; // 5
    x-=y; // 7
    cout<<"O‘zaro almashgan qiymatlar: ";
    cout<<"x="<<x<<', '<<"y="<<y;
    return 0;
}
```

$x$  va  $y$  o‘zgaruvchilarning qiymati sifatida 5 va 7 sonlari kiritilsa, ekranga ularning almashgan natijasi chop etiladi:

O‘zaro almashgan qiymatlar: x=7, y=5.

##### *Amaliy topshiriqlar*

1.  $x^x$  -funksiyasining hosilasi  $a(a>0)$  nuqtada hisoblansin.
2. Berilgan  $a$  soni uchun  $\ln(\ctgx-1)=a$  tenglamaning  $(\pi, 2\pi)$  oraliqdagi yechimi topilsin.
3. Berilgan koeffisient va o‘ng tomon qiymatlarida tenglamalar sistemasining determinanti noldan farqli deb hisoblab sistema yechimi topilsin:

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2. \end{cases}$$

4. Berilgan  $R$  radiusli aylananing uzunligi, doiraning yuzasi va sharning hajmi

hisoblansin.

5. Uchburchak uchlarining koordinatalari asosida uning yuzasi va perimetri topilsin.
6. Berilgan to'rt xonali son raqamlarining ko'paytmasi topilsin.
7. Massalari mos ravishda  $m_1$  va  $m_2$  bo'lgan va bir biridan  $r$  masofada joylashgan ikkita jismning o'zaro tortishish kuchi  $F$  aniqlansin.
8. To'g'riburchakli uchburchakning gipotenuzasi va kateti berilgan bo'lsa, ikkinchi katet va ichki chizilgan aylananing radiusi hisoblansin.
9. Aylana uzunligi berilgan, bu aylana bilan chegaralangan doira yuzasi topilsin.
10. Uchburchak burchaklarining kattaliklari va ichki chizilgan aylana radiusi bilan berilgan. Uchburchak tomonlari uzunliklari hisoblansin.
11. Uchburchak tomonlari berilgan. Quyidagilar aniqlansin:  
a) balandliklari;      b) medianalari;  
e) bissektrisalari;      d) ichki va tashqi chizilgan aylana radiuslari.
12. Butun turdagi  $h$  o'zgaruvchiga berilgan  $k$  ( $k \geq 100$ ) musbat sonning oxiridan uchinchi raqami o'zlashtirilsin (masalan, agar  $k=130985$  bo'lsa,  $h=9$ ).
13. Butun turdagi  $d$  o'zgaruvchiga haqiqiy musbat  $x$  sonining kasr qismining birinchi raqami o'zlashtirilsin (masalan, agar  $x=32.597$  bo'lsa,  $d=5$ ).
14. Agar hozir sutkaning  $k$ -sekundi bo'lsa, sutka boshidan o'tgan aniq vaqt - soat( $h$ ), minut( $m$ ) va sekund( $s$ ) hisoblansin (masalan, agar  $k=13257=3 \cdot 3600 + 40 \cdot 60 + 57$  bo'lsa,  $h=3$ ,  $m=40$  va  $s=57$ ).
15. Soat milining sutka boshidagi holati bilan  $h$  soat,  $m$  minut va  $s$  sekunddagi holatlari orasidagi  $f$  burchak (graduslarda) aniqlansin ( $0 \leq h \leq 11$ ,  $0 \leq m \leq 59$ ).
16. Soat milining sutka boshiga nisbatan  $f$  gradus burilishiga mos keluvchi  $h$  soat,  $m$  minut va  $s$  sekund topilsin ( $0 \leq f \leq 360^\circ$ ,  $f$  - haqiqiy son).
17. Birinchi yanvar kuni dushanba bo'lib, kabisa bo'lmagan biror yilning  $k$ -kuni ( $1 \leq k \leq 365$ ) haftaning qaysi kuniga to'g'ri kelishi aniqlansin va bu qiymat  $n$  butun o'zgaruvchiga o'zlashtirilsin ( $1 \leq n \leq 7$ ).
18. Og'irligi bir kilogramm bo'lgan mahsulotning narxi berilgan. Uning og'irligi grammlarda kiritilsin va to'lash zarur bo'lgan pul miqdori chop etilsin.
19. 10 metr radiusli silindrik shaklga ega bo'lgan suv bosimi minorasidagi suv sathining balandligi berilgan bo'lsa, suvning hajmi hisoblansin.
20. Bolalar bog'chasiga bir oylik to'lov 70000 so'm (bir oy - 22 kun). Agar bola bog'chaga  $N$  ( $0 < N < 23$ ) kun kelmagan bo'lsa, bir oy uchun qancha to'lash kerak bo'ladi?
21.  $R$  radiusli doiraga ichki chizilgan muntazam  $n$ -burchakning perimetri va yuzasi hisoblansin.
22. Turg'un suvdagi qayiq tezligi  $V$  km/s. Daryo suvi oqimining tezligi  $U$  km/s ( $U < V$ ). Qayiq ko'lda  $T_1$  soat, daryoda esa (oqimga qarshi)  $T_2$  soat harakat qilgan. Qayiq suzgan umumiy  $S$  masofa topilsin.

23. Birinchi avtomobil tezligi  $V_1$  km/s, ikkinchisidiki -  $V_2$  km/s, ular orasidagi masofa -  $S$  km. Avtomobillar bir-biridan uzoqlashsa (bir-biriga qarab harakat qilganda),  $T$  soatdan keyin ular orasidagi masofa qanday bo'ladi?
24. Asoslari  $a$  va  $b$  ( $a > b$ ), katta asosdagi burchagi  $\alpha$  bo'lgan teng yonli trapetsiyaning perimetri hamda yuzasi topilsin (burchak radianda beriladi).
25. Noldan farqli berilgan  $R_1, R_2, R_3$  elektr qarshiliklari uchun  $R_0$  hisoblansin. Bunda:  $\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$ .
26. Xodimning oylik ish haqiga 45% mukofot puli qo'shilsin. Hosil bo'lgan miqdordan 17% daromad solig'i, 1,5% kasaba uyushmasi va 1% nafaqa solig'i ushlab qolinsin. Qo'lga tegadigan pul miqdori chop etilsin.
27. Uch xonali butun son ( $k$ ) raqamlari yig'indisini ( $s$ ) butun o'zgaruvchiga o'zlashtirilsin.
28. Teng tomonli uchburchak tomoni berilgan, uchburchak yuzasi topilsin.
29. Uchta musbat son berilgan. Sonlar o'rta geometrigining kasr qismi topilsin.
30. Berilgan katetlari bo'yicha to'g'ri burchakli uchburchakning perimetri va yuzasi hisoblansin.
31. Berilgan ikki tomoni va ular orasidagi burchak (gradusda) asosida uchburchakning uchinchi tomoni va yuzasi topilsin.
32. Berilgan uch xonali son raqamlarini teskari tartibda yozishdan hosil bo'lgan son topilsin. Masalan, 345 sonining teskari tartibi 543 bo'ladi.

#### 4. Mantiqiy ifodalar

##### *Namunaviy masala*

Berilgan uch xonali butun sonning raqamlari o'zaro teng yoki teng emasligi aniqlansin.

##### *Yechish usuli*

Masala C++ tilining butun sonlar ustidagi arifmetik amallardan foydalangan holda yechiladi. Berilgan butun  $a$  va  $b$  sonlar uchun '/' amali  $a/b$  bo'linmaning butun qismini, '%' amali  $a \% b$  bo'linmaning butun qoldiqini beradi. Bu bo'lishlardan foydalanib, berilgan sonning raqamlarini ajratib olish va ularni o'zaro solishtirish mumkin.

##### *Programma matni*

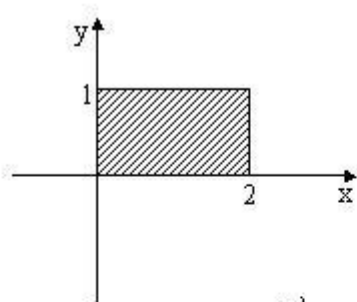
```
#include <iostream.h>
int main()
{
    unsigned int n,a2,a1,a0; // n=a2a1a0 ko'rinishida
    cout<<"\nn - qiymatini kiriting: ";
    cin>>n;
    if(n<100||n>999)
```

```
{  
    cout<<"Kiritilgan son 3 xonali emas!";  
    return 1;  
}  
a2=n/100;  
a1=(n%100)/10;  
a0=n%10;  
cout<<"Berilgan son raqamalri o'zaro teng";  
if(a2!=a1 && a1!=a0 && a2!=a0) cout<<"!";  
else cout<<" emas!";  
return 0;  
}
```

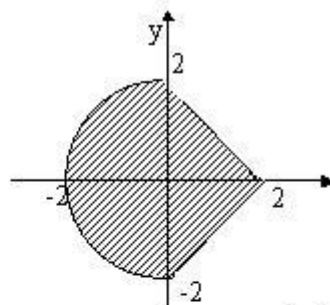
### *Amaliy topshiriqlar*

1. Ifodaning qiymati topilsin:
  - a) agar  $x=0.3$ ,  $y=-1.6$  bo'lsa  $x^2 + 2y \leq 4$ ;
  - b) agar  $k=15$  bo'lsa,  $k\%7=k/5-1$ ;
  - d) agar  $p=0.182$  bo'lsa,  $(10-p)\%2=0$ .
2. Quyidagi shartlar bajarilganda "rost", aks holda "yolg'on" qiymat qabul qiluvchi mantiqiy munosabatlar C++ tilida yozilsin:
  - a)  $k$  butun soni 7 ga bo'linadi;
  - b)  $ax^2+bx+c=0$  tenglama haqiqiy ildizlarga ega emas;
  - d)  $(x,y)$  nuqta, markazi  $(1,0)$  nuqtada bo'lgan  $r$  radiusli aylananing tashqarisida yotadi;
  - e)  $n$  natural soni – to'liq kvadrat.
3. Ifodalarning qiymatlari hisoblansin:
  - a)  $\neg \left\{ \frac{n}{2} \right\}$ , agar  $n=15$ . Bu yerda " $\{ \}$ " – bo'linma qoldig'i;
  - b)  $t \wedge \left\{ \frac{n}{2} \right\} = 0$ , agar  $t=true$ ,  $n=101010$ ;
  - d)  $(x+y \neq 0) \wedge (y > x)$ , agar  $x=2$ ,  $y=1$ ;
  - e)  $(x+y \neq 0) \vee (y > x)$ , agar  $x=2$ ,  $y=1$ ;
  - f)  $a \vee \neg b$ , agar  $a=false$ ,  $b=true$ .
4. Quyidagi shartlar bajarilganda rost, aks holda yolg'on qiymat qabul qiluvchi mantiqiy munosabatlar C++ tilida yozilsin:
  - a)  $0 < x < 1$ ;
  - b)  $x = \max(x, y, z)$ ;
  - d)  $x \neq \max(x, y, z)$  (inkor amali qo'llanilmasin);
  - e)  $a, b$  mantiqiy o'zgaruvchilardan kamida bittasi true;
  - f) har ikkala  $a, b$  mantiqiy o'zgaruvchilar qiymatlari true.
5. Ayniyatlar isbotlansin:
  - a)  $a \wedge \neg a \equiv false$ ;      b)  $a \vee \neg a \equiv true$ ;      d)  $\neg \neg a \equiv a$ ;
  - e)  $true \vee a \equiv true$ ;      f)  $false \wedge a \equiv false$ ;      g)  $a \vee a \equiv a$ .
6. Hisoblansin:
  - a)  $false \vee (1/1 > 0)$ ;      b)  $(1/2 > 0) \wedge true$ .
7. Ifodalardagi amallar bajarilish tartibi ko'rsatilsin:
  - a)  $a \& \& b \parallel !c \& \& d$ ;      b)  $(x > 0) \parallel t \& \& x \% 2 == 0 \parallel (y * y != 4)$ .
8. Mantiqiy  $a$  va  $b$  o'zgaruvchilarning  $a=true$  va  $b=false$  qiymatlari uchun quyidagi ifodalar hisoblansin:
  - a)  $a \parallel b \& \& !a$ ;      b)  $(a \parallel b) \& \& !a$ ;      d)  $!a \& \& b$ ;      e)  $!(a \& \& b)$ .
9. Quyidagi shartlar bajarilganda rost, aks holda yolg'on qiymat qabul qiluvchi ifodalar C++ tilida yozilsin:
  - a)  $x \in [0,1]$ ;      b)  $x \notin [0,1]$ ;      d)  $x \in [2,5] \vee x \in [-1,1]$ ;      e)  $x \notin [2,5] \vee x \in [-1,1]$ .

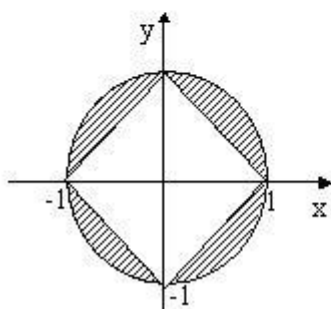
- f)  $x, y, z$  sonlaridan har biri musbat;  
 g)  $x, y, z$  sonlaridan hech bo'lmaganda biri musbat;  
 h)  $x, y, z$  sonlaridan hech biri musbat emas;  
 i)  $x, y, z$  sonlaridan faqat biri musbat;  
 j) mantiqiy o'zgaruvchi  $a$  true,  $b$  esa false qiymatini qabul qilgan holda;  
 k)  $y$ - yil kabisa yili (Kabisa yili 4 ga karrali yillar hisoblanadi. Biroq, 100 ga karrali yillar orasida faqat 400 ga karrali yillar kabisa yili deyiladi. Masalan, 1700, 1800, 1900-oddiy yillar, 2000-kabisa yili).
10. Agar  $(x, y)$  nuqta bo'yalgan sohaga tegishli bo'lsa,  $t$  mantiqiy o'zgaruvchi true qiymatini qabul qiladigan ifoda yozilsin (rasmga qarang).



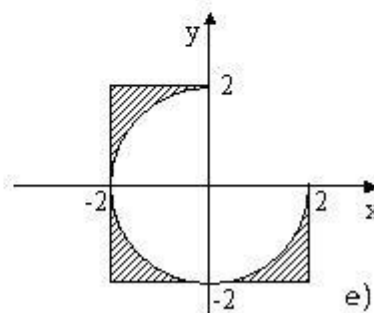
a)



b)



d)



e)

11. Quyidagi shartlarga mos keluvchi soha tekislikda tasvirlansin:  
 a)  $(y > x) \& \& (y + x > 0) \& \& (y \leq 1)$ ;  
 b)  $(x * x + y * y < 1) || (y > 0) \& \& (y \leq 1)$ .
12. Ifodaning qiymati hisoblansin:  
 a)  $\text{false} < \text{true}$ ; b)  $(32 || \text{false}) == 1$ ; d)  $9 + 3 * \text{true}$ ; e)  $16 + \text{true} / 2$ .
13. Ifodaning qiymati hisoblansin:  
 a)  $! (--s) \& \& ((\text{int})(s) == 1)$ , agar  $s = \text{true}$ ;  
 b)  $(p < \text{true}) == (q == \text{false})$ , agar  $p = q = \text{true}$ ;  
 d)  $a \& \& b > a || b$ , agar  $a = \text{true}$ ,  $b = \text{false}$ .
14. Quyidagi shartlar bajarilganda rost, aks holda yolg'on qiymat qabul qiladigan ifoda C++ tilida yozilsin:  
 a) butun  $n$  va  $m$  sonlari bir paytda toq yoki juft sonlar;  
 b)  $a, b$  mantiqiy o'zgaruvchilardan faqat bittasi true qiymatiga ega;  
 d)  $a, b, c$  mantiqiy o'zgaruvchilardan faqat bittasi true qiymatini qabul qiladi.
15. Quyidagi ifodalar o'rinli bo'ladigan soha XOY tekisligida chizilsin:  
 a)  $(\text{fabs}(x) \leq 1) == (\text{fabs}(y) > 1)$ ;



- b)  $(x * x + y * y <= 4) == (y <= x)$ .
16. Ayniyatlar isbotlansin:
- a)  $!(a || b) == (!a) \&\& (!b)$ ;
- b)  $a \&\& (b || c) == (a \&\& b) || (a \&\& c)$ ;
- d)  $a <= b == !a || b$ ;
- e)  $a \&\& b == (a < true) < b$ ;
- f)  $!a == a < true$ .
17. Mantiqiy  $a, b, c$  o'zgaruvchilar uchun quyidagi ifodalarni taqqoslash amallari qatnashmagan ko'rinishga keltirilsin:
- a)  $a < b$ ;                      b)  $a == b$ ;                      d)  $(a < b) == a$ .
18. Agar  $a = true$  va  $x = 1$  bo'lsa, quyidagi mantiqiy  $d$  o'zgaruvchi qanday qiymat qabul qiladi?
- a)  $d = x < 2$ ;                      b)  $d = !a || x \% 2$ ;                      d)  $d = a \% 2 != x$ .
19. Quyidagi shart bajarilganda  $t$  mantiqiy o'zgaruvchisi  $true$ , aks holda  $false$  qiymatini o'zlashtirsin:
- a)  $x, y, z$  sonlar o'zaro teng;
- b)  $x, y, z$  sonlardan faqat ikkitasi o'zaro teng;
- d)  $x$  musbat son;
- e)  $p$  soni  $q$  ga qoldiqsiz bo'linadi ( $p$  va  $q$ -natural sonlar);
- f)  $ax^2 + bx + c = 0$  tenglama bitta yechimga ega, bu yerda  $a, b$  va  $c$  o'zgaruvchilar 0 bo'lishi mumkin;
- g) uch xonali butun  $k$  sonining o'nli yozuviga '5' raqami kiradi;
20. Shaxmat taxtasining kataklari  $(x, y)$  butun sonlar juftligi bilan beriladi ( $1 \leq x, y \leq 8$ ). Berilgan  $(h_1, v_1)$  va  $(h_2, v_2)$  kataklar uchun quyidagi mulohazalar mantiqiy ifoda ko'rinishida yozlsin:
- a) kataklari bir xil rangga ega;
- b)  $(h_1, v_1)$  katagida joylashgan «ot»  $(h_2, v_2)$  katakka xavf soladi;
- d)  $(h_1, v_1)$  katagida joylashgan «farzin»  $(h_2, v_2)$  katakka xavf soladi.

## 6. '?' shart amali

Quyidagi misollar '?' shart operatori yordamida yechilsin. Berilgan haqiqiy  $x$  soni uchun  $y$  hisoblansin.

### Namunaviy masala

Berilgan haqiqiy  $x$  soni uchun  $y$  qiymati hisoblansin.

$$y = \begin{cases} x^2, & \text{agar } -3 \leq x < 3; \\ x^2 + 4x, & \text{agar } x < -3; \\ 3 - x, & \text{aks holda.} \end{cases}$$

### Yechish usuli

Hisoblash shartlari nisbatan sodda bo'lganligi sababli interval ko'rinishida berilgan funksiya qiymatini ichma-ich joylashgan '?' shart amalini qo'llash orqali echiladi.

### **Programma matni**

```
#include<iostream.h>
#include<math.h>
int main()
{
    float y,x;
    cout<<"x="; cin>>x;
    y=(x>=3)?3-x: // x>=3 bo'lsa y=3-x aks holda
    (x>=-3&&3>x)?x*x:x*x+4*x; // agar -3<=x<3 bo'lsa y=x*x
    // aks holda y=x*x+4*x;

    cout<<"y="<<y;
    return 0;
}
```

### **Amaliy moniuriqlar**

$$1. \quad y = \begin{cases} 5x^3 + 6x^2 - 2x, & |x| \leq 1; \\ \frac{x+3}{|x|}, & x > 1; \\ 0, & x < -1. \end{cases}$$

$$2. \quad y = \begin{cases} x^2, & -2 \leq x < 2; \\ x^2 + 4x + 5, & x < -2; \\ 2 - x, & x \geq 2. \end{cases}$$

$$3. \quad y = \begin{cases} 0, & x \leq 0; \\ x, & 0 < x \leq 1; \\ x^4, & x > 1. \end{cases}$$

$$4. \quad y = \begin{cases} \sin x + \sqrt{|x-5|}, & x < 5; \\ 5.45^2 \cos t + \ln x, & x = 5; \\ (x-5)^2 \operatorname{tg} \frac{x}{2}, & x > 5. \end{cases}$$

$$5. \quad y = \begin{cases} -\frac{1}{x^2}, & x \leq -1; \\ x^2, & -1 < x \leq 2; \\ 4, & x > 2. \end{cases}$$

$$6. \quad y = \begin{cases} \operatorname{arctg} \sqrt{|x-2|}, & x < -1; \\ 2.7^t \cos x, & -1 \leq x \leq 1; \\ \ln(x-1) \sin \frac{\pi x}{2}, & x > 1. \end{cases}$$

$$7. \quad y = \begin{cases} \operatorname{ctg} x + 3.215 \sqrt{|x+2|}, & x < -2; \\ 1385^2 \cos t, & -2 \leq x \leq 5; \\ (x-2)^{\frac{2}{3}} \sin \frac{\pi x}{2}, & x > 5. \end{cases}$$

$$8. \quad y = \begin{cases} \sin x + \sqrt{|x+2|}, & x < -2; \\ x^2 \cos t + \ln(x+6), & -2 \leq x \leq 0; \\ (x+5)^{\frac{1}{3}} \operatorname{tg} \frac{x}{2}, & x > 2. \end{cases}$$

$$9. \quad y = \begin{cases} \operatorname{tg} x + \sqrt{1+|x|}, & x < -5; \\ 6.78^{\frac{3}{4}} + 2x + \sin x, & -5 \leq x \leq 2; \\ 3x + \sin x, & x > 2. \end{cases}$$

$$10. \quad y = \begin{cases} \operatorname{ctg} x + \sqrt{1+|x-2|}, & x < 0; \\ (5.12+x)^{\frac{1}{3}} + \sin t, & 0 \leq x \leq 1; \\ 3x + \sin x, & x > 1. \end{cases}$$

$$11. y = \begin{cases} x \operatorname{tg} x + \sqrt{\ln|x-2|}, & x < -2; \\ (x^2 + 3)x, & -2 \leq x \leq 2; \\ (x-2)\sin\frac{\pi x}{2}, & x > 2. \end{cases}$$

$$13. a = 1.5;$$

$$y = \begin{cases} \operatorname{tg} x + \sqrt{\ln|a-3|}, & x < -3; \\ (a^2 - 3)^2 - \sin 2x, & -3 \leq x \leq 3; \\ (a+3) - \cos \pi x, & x > 3. \end{cases}$$

$$15. a = 2; b = .5;$$

$$y = \begin{cases} 1, & x < 1; \\ a x^2 \ln x, & 1 \leq x \leq 2; \\ e^{ax} \cos x, & x > 2. \end{cases}$$

$$17. a = 2.8; b = -0.3; c = 4;$$

$$y = \begin{cases} a x^2 + b x + c, & x < 1.2; \\ a/x + \sqrt{x^2 - 1}, & x = 1.2; \\ (a + b x)/\sqrt{x^2 + 1}, & x > 1.2. \end{cases}$$

$$19. y = \begin{cases} 1.5 \cos^2 x, & x < 1; \\ (x-2)^2 + 6, & 1 \leq x \leq 2; \\ 3 \operatorname{tg} x, & x > 2. \end{cases}$$

$$12. y = \begin{cases} \operatorname{ctg} x + \sqrt{3 + |x|}, & x < -5; \\ (3x+1)^2 + \sin \pi, & -5 \leq x \leq 3; \\ (x-3) + \sin \pi x, & x > 3. \end{cases}$$

$$14. a = 1.5;$$

$$y = \begin{cases} \pi x^2 - 7/x^2, & x < 1.3; \\ a x^3 + 7\sqrt{x}, & x = 1.3; \\ \operatorname{tg}(x + 7\sqrt{x}), & x > 1.3 \end{cases}$$

$$16. a = 1.65; b = 1.1$$

$$y = \begin{cases} \pi x^2 - 7/x^2, & x < 1.4; \\ a x^3 + 7\sqrt{x^2 - 1}, & x = 1.4; \\ (a + b x)/\sqrt{x^2 + 1}, & x > 1.4. \end{cases}$$

$$18. a = 2; b = 0.5;$$

$$y = \begin{cases} 1, & x < 1; \\ a x^2 \ln x, & 1 \leq x \leq 2; \\ e^{ax} \cos x, & x > 2. \end{cases}$$

$$20. y = \begin{cases} x/n, & 0 \leq x \leq n; \\ n^2 \left( \frac{x}{2} - \sin x \right), & x > n; \\ \cos(x), & x < 0. \end{cases} \quad n = 10;$$

## 7. if-else shart operatori

### Namunaviy masala

Haqiqiy turdagi  $x_1, y_1, x_2, y_2, x_3, y_3$  sonlari berilgan. Koordinata markazi uchlari  $(x_1, y_1), (x_2, y_2)$  va  $(x_3, y_3)$  nuqtalarda bo'lgan uchburchakka tegishlimi?

### Yechish usuli

Birorta  $(x, y)$  nuqta uchburchak ichiga tegishli bo'ladi, agar bu nuqtaning uchburchak uchlari bilan hosil qiluvchi uchburchaklari yuzalarining ( $s_1, s_2$  va  $s_3$ ) yig'indisi shu uchburchak yuzasiga ( $s$ ) teng bo'lsa:  $s = s_1 + s_2 + s_3$ . Shuni qayd etish kerakki, ikkita haqiqiy turdagi qiymatlarning o'zaro tengligini tekshirish ular ayirmasining absolyut qiymati berilgan aniqlikdan kichikligi orqali aniqlanadi.

Uchburchakning yuzasi Geron formulasi yordamida topiladi:

$$s = \sqrt{p(p-a)(p-b)(p-c)},$$

bu yerda  $a, b, c$  - uchburchak tomonlarining uzunliklari,  $p$  - uchburchak perimetrining yarmi.

***Programma matni***

```
#include <iostream.h>
#include <math.h>
int main()
{
    float x1,y1, x2,y2, x3,y3; // uchburchak uchlarining koordinatalari
    float a,b,c ;              // uchburchak tomonlari
    float s1,s2,s3;            // ichki uchburchaklar yuzalari
    float s;                    // uchburchak yuzasi
    float p;                    // uchburchak perimetrining yarmi
    const float eps=0.0001;    // tenglik aniqligi
    cout<<"\n1-nuqta koordinatasini kiriting(x1,y1): ";
    cin>>x1>>y1;
    cout<<"\n2-nuqta koordinatasini kiriting(x2,y2): ";
    cin>>x2>>y2;
    cout<<"\n3-nuqta koordinatasini kiriting(x3,y3): ";
    cin>>x3>>y3;
    //{(x1,y1),(x2,y2),(x3,y3)} uchburchak yuzasini hisoblash(s)
    a=sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
    b=sqrt((x2-x3)*(x2-x3)+(y2-y3)*(y2-y3));
    c=sqrt(pow(x1-x3,2)+pow(y1-y3,2));
    p=(a+b+c)/2;
    s=sqrt(p*(p-a)*(p-b)*(p-c));
    //{(0,0),(x1,y1),(x3,y3)} uchburchak yuzasini hisoblash (s1)
    a=sqrt(x1*x1+y1*y1);
    b=sqrt(x3*x3+y3*y3);
    p=(a+b+c)/2;
    s1=sqrt(p*(p-a)*(p-b)*(p-c));
    //{(0,0),(x2,y2),(x3,y3)} uchburchak yuzasini hisoblash (s2)
    a=sqrt(x2*x2+y2*y2);
    c=sqrt(pow(x2-x3,2)+pow(y2-y3,2));
    p=(a+b+c)/2;
    s2=sqrt(p*(p-a)*(p-b)*(p-c));
    //{(0,0),(x1,y1),(x2,y2)} uchburchak yuzasini hisoblash (s3)
    b=sqrt(x1*x1+y1*y1);
    c=sqrt(pow(x2-x1,2)+pow(y2-y1,2));
    p=(a+b+c)/2;
    s3=sqrt(p*(p-a)*(p-b)*(p-c));
    // s=s1+s2+s3 shartni tekshirish. Bunda tenglikka tekshirish
    // qiymatlarni ayirmasining absolyut qiymatini nolga
    // yaqinligi bilan almashtiriladi, chunki haqiqiy sonlarni
    // ustida amallar bajarilganda aniqlik yo'qotilishi mumkin
```

```

if(fabs(s-(s1+s2+s3))<eps)
cout<<"Koordinata markazi uchburchak ichida.";
else
cout<<"Koordinata markazi uchburchak ichida emas.";
return 0;
}

```

Programma bajarilganda koordinatalari  $(-4,-1),(4,-3),(2,3)$  bo'lgan uchburchak kiritilsa, ekranga

Koordinatalar markazi uchburchak ichida.

xabari chop etiladi.

### *Amaliy topshriqlar*

1. Agar tomonlarining uzunliklari ixtiyoriy  $a$ ,  $b$  va  $c$  sonlarga teng bo'lgan uchburchakni qurish mumkin bo'lmasa 0, aks holda – uchburchak teng tomonli bo'lsa 3, teng yonli bo'lsa 2 va boshqa hollar uchun 1 qiymatini chop qiluvchi programma tuzilsin.
2. Agar uchta haqiqiy, o'zaro teng bo'lmagan  $x, y$  va  $z$  sonlar yig'indisi 1 dan kichik bo'lsa, uchta sonning eng kichigi qolganlari yig'indisining yarmisi bilan almashtirilsin, aks holda  $x$  va  $y$  lardan kichigi qolganlari yig'indisining yarmi bilan almashtirilsin.
3. Berilgan 50 ta haqiqiy sonlarning eng kattasini topadigan programma tuzilsin.
4. Haqiqiy  $x, y$  va  $z$  sonlar berilgan bo'lsa, quyidagilar aniqlansin:  
 a)  $\max(x, y, z)$ ;                      b)  $\max(x, y) + \min(y, z)$ ;  
 d)  $\max(x+y+z, x*y*z)$ ;      e)  $\min((x+y+z)/2, x*z+1)$ .
5. Uchta  $x$ ,  $y$  va  $z$  haqiqiy sonlar berilgan, agar ular monoton bo'lsa ularning qiymatlari ikkilantirilsin, aks holda har bir o'zgaruvchining ishorasi qarama-qarshisiga almashtirilsin.
6. Butun  $n$  ( $n>0$ ) va  $n$  ta haqiqiy sonlar berilgan. Ular orasidan manfiylari nechtaligi aniqlansin.
7.  $OX$  va  $OY$  o'qlarida yotmaydigan nuqta koordinatalari bilan berilgan. Bu nuqta joylashgan koordinata choragi aniqlansin.
8. Bo'sh bo'lmagan va oxiri 0 soni bilan tugaydigan musbat butun sonlar ketma-ketligi berilgan (0 soni ketma-ketlikka kirmaydi va uning tugaganligini bildiradi). Ketma-ketlikning o'rta geometrik qiymati hisoblansin.
9. Haqiqiy  $x, y$  va  $z$  sonlari berilgan bo'lib,  $x < y < z$  munosabat o'rinli bo'lsa bu sonlar ikkilantirilsin, aks holda bu sonlar absolyut qiymatlari bilan almashtirilsin.
10. Uchta ixtiyoriy  $a, b$  va  $c$  son berilgan. Tomonlarining uzunliklari shu sonlarga teng bo'lgan uchburchak mavjudmi?
11. Sonlar o'qida uchta  $A$ ,  $B$  va  $C$  nuqtalar joylashgan.  $B$  va  $C$  nuqtalardan qaysi biri  $A$  nuqtaga yaqin masofada joylashgan bo'lsa, shu masofa chop

etilsin.

12. Berilgan uch xonali son raqamlari orasida bir xillari bor yoki yo'qligi aniqlansin?
13. Berilgan  $x$  uchun quyidagi ifodalarning qiymatlari o'sish tartibida chop etilsin:  $chx$ ,  $1+|x|$  va  $(1+x^2)^x$ .
14.  $a_1x+b_1y=c_1$  va  $a_2x+b_2y=c_2$  tenglamalar bilan berilgan chiziqlarning kesishish nuqtasi koordinatalarini chop etadigan yoki bu chiziqlarning ustma-ust tushishligi, yoki paralleligi haqida ma'lumot beradigan programma tuzilsin. Bu yerda  $a_1, b_1, c_1, a_2, b_2$  va  $c_2$  – berilgan sonlar.
15.  $ax^4+bx^2+c=0$  tenglamaning haqiqiy ildizlarini topadigan yoki ildizi yo'qligi haqida ma'lumot beradigan programma tuzilsin.
16. Shaxmat taxtasidagi maydonlar sakkizdan katta bo'lmagan sonlar juftligi bilan aniqlanadi: birinchi son shaxmat taxtasi maydonining vertikal nomeri (chapdan o'nga), ikkinchisi – gorizontal nomeri (pastdan yuqoriga). Sakkizdan katta bo'lmagan  $k$ ,  $l$ ,  $m$  va  $n$  sonlari berilgan. Quyidagi masalalar yechilsin:
  - a)  $(k,l)$  maydonidagi ruh bir yurishda  $(m,n)$  maydoniga o'tishi mumkinmi? Agar mumkin bo'lmasa, ikkita yurishda o'tish yo'llari ko'rsatilsin;
  - b)  $(k,l)$  maydonidagi farzin bir yurishda  $(m,n)$  maydoniga o'tishi mumkinmi? Agar mumkin bo'lmasa, ikkita yurishda o'tish yo'llari ko'rsatilsin.
17. Uchburchak uchlarining koordinatalari  $M_1(x_1, y_1)$ ,  $M_2(x_2, y_2)$  va  $M_3(x_3, y_3)$  berilgan. Berilgan  $M(x, y)$  nuqta uchburchak ichida yotish yoki yotmasligi aniqlansin.
18. Berilgan  $a_1, a_2, a_3$  va  $a_4$  butun sonlar ichida uchtasi bir-biriga teng. Boshqalaridan farqli bo'lgan sonning tartib nomeri chop etilsin.
19. Butun turdagi  $a$ ,  $b$  va  $c$  o'zgaruvchilar qiymati shunday almashtirilsinki, natijada  $a \leq b \leq c$  munosabat o'rinli bo'lsin.
20. Natural  $n(n \leq 9999)$  soni berilgan. Sonni to'rt xonali deb hisobga olgan holda ushbu sonning palindrom ekanligi aniqlansin (chapdan va o'ngdan bir xil o'qiladigan sonlar, masalan, 1221, 5555, 440 sonlari palindrom sonlar hisoblanadi).
21. O'lchamlari  $a \times b \times c$  bo'lgan to'g'ri burchakli qutiga  $p \times r \times q$  o'lchamdagi to'g'ri burchakli taxta bo'lagini joylashtirish mumkinmi?
22. Radiusi  $r$  bo'lgan doira ko'rinishidagi alyumin plastinkadan o'lchamlari  $a \times b$  va  $p \times q$  bo'lgan ikkita to'g'ri to'rtburchak shaklidagi plastinkalarni qirqib olish mumkin yoki yo'qligi aniqlansin.
23. Tomonlari koordinata o'qlariga parallel (perpendikulyar) bo'lgan ikkita kvadratlar bosh diagonallarining koordinatalari bilan berilgan:  $(x_1, y_1)$  va  $(x_2, y_2)$  - birinchi kvadrat;  $(x_3, y_3)$  va  $(x_4, y_4)$  - ikkinchi kvadrat. Agar kvadratlar o'zaro kesishmasa 0, urunsa 1 va kesishsa 2 qiymati chop

etilsin.

24. Ikki xonali sonlar ketma-ketligining (1011121314..9899) k - o‘rindagi ( $1 \leq k \leq 180$ ) raqami aniqlansin.
25. 10 sonining darajalaridan tuzilgan ketma-ketlikning (101001000...) k - o‘rindagi raqami aniqlansin.
26. Tekislikdagi nuqta butun sonli koordinatalar bilan berilgan, agar nuqta koordinata o‘qlarida yotmasa 0, agar nuqta (0,0) bilan ustma – ust tushsa 1, agar nuqta OX yoki OY o‘qlarida yotsa, mos ravishda 2 yoki 3 sonlari chop etilsin.
27. Berilgan yilga mos keluvchi asr nomeri chop etilsin. Bunda quidagi holat inobatga olinsin: masalan, 21 asr boshi 2001 yildan hisoblanadi.
28. Qiymati [-999,999] oraliqda yotuvchi butun son berilgan. Son qiymatiga mos ravishda “*manfiy ikki xonali son*”, “*nol soni*”, “*uch xonali musbat son*” kabi satrlar chop qilinsin.
29. Qiymati  $1 \leq x \leq 9999$  bo‘lgan x butun soni berilgan. Bu sonning qiymatiga mos ravishda quydagi satrlar chop etilsin: “*to‘rt xonali juft son*”, “*ikki xonali toq son*” va hokazo.
30. Berilgan p, a va b ( $a < b$ ) sonlar uchun  $\arctan\left(\frac{b-a}{p}\right) = \sqrt{2}$  tenglamaning ildizi  $[a, b]$  kesmaga tegishliligi aniqlansin.
31. Berilgan to‘rt xonali sonning boshidagi ikkita raqamlari yig‘indisi qolgan raqamlari yig‘indisiga teng yoki yo‘qligi aniqlansin.
32. Berilgan haqiqiy musbat son kasr qismining boshidagi uchta raqamlari orasida 0 raqami bormi?

## 9. switch tarmoqlanish operatori

### *Namunaviy masala*

Butun turdagi ixtiyoriy ikkita son berilgan. Bu sonlar ustida tanlangan arifmetik amalga (+, -, \*, /) mos hisoblash amali bajarilsin. Agar arifmetik amallardan farqli belgi kiritilsa, bu haqda xabar berilsin va amalni qayta tanlash taklif qilinsin.

### *Yechish usuli*

Butun a va b sonlari klaviaturadan kiritiladi. Keyin foydalanuvchiga arifmetik amallarni tanlash taklif etiladi. Tanlangan amal amal o‘zgaruvchisiga o‘zlashtiriladi. switch operatori yordamida qaysi amal ekanligi aniqlanib, sonlar o‘rtasida mos amal bajariladi, break operatori bajarilishi orqali tanlash operatoridan tashqariga chiqiladi va natija chop etiladi.

### *Programma matni*

```
#include<iostream.h>
#include<math.h>
int main()
{
    int a,b;
```

```

float natija;
char amal;
cout<<"a="; cin>>a;
cout<<"b="; cin>>b;
nishon:
cout<<"Arifmetik amallar:";
cout<<"+: c=a+b";
cout<<"- : c=a-b";
cout<<"* : c=a*b";
cout<<" / : c=a/b";
cout<<"Amalni tanlang(+,-,*,/): ";
cin>>amal;
switch(amal)
{
    case '+': natija=a+b;break;
    case '-': natija=a-b;break;
    case '*': natija=a*b;break;
    case '/': natija=a/b;break;
    default : cout<<"bunday amal yoq";
    goto nishon;
}
cout<<"Hisoblash natijasi="<<natija;
return 0;
}

```

### ***Amaliy topshiriqlar***

1. Berilganlar turi va o'zgaruvchilar quyidagicha aniqlangan:  
enum Oy{yan,fev,mar,apr,may,iyn,iyl,avg,sen,okt,noy,dek};  
int d1,d2; Oy m1,m2; bool t;  
Agar d1, m1 sana (yil hisobida) d2, m2 sanadan oldin kelsa, t o'zgaruvchiga true, aks holda false qiymat berilsin.
2. Oy m,m1; (Oy turi yuqoridagi masalada berilgan).  
int k,n;  
Quyidagi shart bo'yicha m1 o'zgaruvchiga qiymat berilsin:  
a) m oydan keyingi oyning nomi (dekabrdan keyin yanvar kelishini hisobga olgan holda);  
b) m oydan keyingi k-oyning nomi;  
d) yilning n - oy nomi berilsin.
3. enum Nota{do,re,mi,fa,sol,lya,si};  
enum Oraliq{secund,tersia,qvart,kvint,sekt,septima};  
Nota n1,n2; Oraliq i;  
Berilgan n1 va n2 ( $n1 \neq n2$ ) notalardan tashkil topgan i-oraliq aniqlansin;  
secund - bu ikkita qo'shni (aylana bo'ylab) notalardan tashkil topgan oraliq (masalan, re va mi, si va do), tersia – bu bitta notadan keyingi oraliq (masalan, fa va lya, si va re) va hokazo.



4. enum Mavsum {qish,bahor,yoz,kuz};  
Oy m; Mavsum s; (Oy turi 1-masalada aniqlangan).  
Berilgan m oyga mos keluvchi s mavsum aniqlansin.
5. enum Davlat{Germaniya,Quba,Laos,Monaqo,Nepal,Polsha};  
enum Qita{Osiyo,Amerika,Evropa};  
Davlat davlat; Qita qita;  
Davlatning nomi bo'yicha u joylashgan qit'a nomi aniqlansin.
6. enum Birlik{desimetr,kilometr,metr,millimetr,santimetr};  
float x; Birlik r;  
Berilgan r birlikdagi x o'zgaruvchining qiymati metrlarda aniqlansin.
7. Berilgan k o'zgaruvchi qiymati ( $0 \leq k \leq 15$ ) rim raqamlari ko'rinishida chop qilinsin.
8. enum Kelishik{bosh,qar,tush,jun,ur\_payt,chiq};  
enum Suz{ruchka,qalam,daftar,eshik};  
Suz s; Kelishik k;  
Berilgan s so'zni k kelishikda chop qilinsin. Masalan, s=daftar va k=jun bo'lganda "daftarga" so'zi chop qilinsin.
9. enum Yunalish{shimol,sharq,janub,garb};  
enum Buyruq{oldinga,unga,orqaga,chapga};  
Yunalish k1,k2;  
Buyruq br;  
Kema avvaliga k1 yo'nalish bo'yicha ketayotgan edi, keyin uning yo'nalishi br buyruqqa asosan o'zgartirildi. Kemaning yangi k2 yo'nalishi aniqlansin.
10. Oy oy; (1-masalaga qaralsin).  
int kun;  
Berilgan oydagi kunlar soni kun o'zgaruvchiga o'zlashtirilsin (yil kabisa yili emas deb hisoblansin).
11. int yil,kun;Oy oy; (Oy turi 1-masalada aniqlangan).  
bool t;  
Agar yil, oy, kun uchlik to'g'ri sanani aniqlasa, t o'zgaruvchiga true, aks holda false qiymat berilsin (masalan, 31 iyun va hokazolarda false).
12. int yil,yil1,kun,kun1;  
Oy oy,oy1; (Oy turi 1-masalada aniqlangan).  
Berilgan yil, oy, kun sanasi bo'yicha keyingi kun sanasi – yil1,oy1,kun1 aniqlansin.
13. int yil\_kuni,oy\_kuni;  
Oy oy; (Oy turi 1-masalada aniqlangan).  
a) Kabisa yilning oy, oy\_kuni sanasiga mos keluvchi kunning yildagi tartib nomeri yil\_kuni aniqlansin.  
b) Kabisa yilining hisobi bo'yicha yil kuniga mos keluvchi oy sanasi aniqlansin.
14. enum Hafta\_Kuni{yaksh,dush,sesh,chor,paysh,juma,shanba};  
int kun,k13; Oy oy; (Oy turi 1-masalada aniqlangan).

Hafta\_Kuni h\_kun1,h\_kun2;

Agar yil kabisa yili bo'lmasa, va uning 1 yanvari haftaning h\_kun1 kuniga to'g'ri kelsa, quyidagilar aniqlansin:

a) kun, oy sanaga mos keluvchi haftaning h\_kun2- kuni;

b) yildagi oyning 13-kuniga mos keluvchi dushanba kunlarining k-soni.

15. Eski yapon kalendarida 60 yillik takrorlanish qabul qilingan va bu takrorlanish o'z navbatida beshta 12 yillik takrorlanish ostilaridan (qismlaridan) iborat. Qism takrorlanishlar quyidagi ranglarning nomi bilan belgilangan: yashil, qizil, sariq, oq va qora. Har bir takrorlanish ostining ichidagi yillar hayvonlarning nomi bilan belgilangan: sichqon, sigir, yo'lbars, quyon, ajdarho, ilon, ot, qo'y, maymun, tovuq, it va to'ng'iz (1984-yil – yashil sichqon yili – keyingi takrorlanishning boshi bo'lgan). Eramizning biror yili kiritilib, uning eski yapon kalendaridagi nomi chop qilinsin.

## 10. for takrorlash operatori

### *Namunaviy masala*

Fibonachchi sonlari quyidagicha aniqlanadi:  $f_0 = f_1 = 1, f_n = f_{n-1} + f_{n-2}$ ,  $n=2,3,\dots$ . Fibonachchi sonlaridan hosil bo'lgan ketma-ketlikning 40 - hadi topilsin.

### *Yechish usuli*

Fibonachchi sonlari ketma-ketligida har bir hadi (2-hadidan boshlab) o'zidan oldingi ikkita hadning yig'indisiga teng. Shu sababli, ketma-ketlikni hosil qilish uchun uchta o'zgaruvchi etarli:  $f_0, f_1, f_2$ . Ketma-ketlikning 2-hadi  $f_2 = f_0 + f_1$  ko'rinishda hisoblanadi, 3-hadni topish uchun esa  $f_0 = f_1, f_1 = f_2$  amallari bajarilib, yana  $f_2 = f_0 + f_1$  topiladi. Bu yerda garchi  $f_2$  qiymat hosil bo'lsa ham, amalda u  $f_3$  qiymatiga teng bo'ladi. Jarayon 39 marta takrorlannib,  $f_{40}$ - had topiladi.

### *Programma matni*

```
#include<iostream.h>
int main()
{
    const int n=40;
    int f0,f1,f2,i;
    f0=f1=1;
    for(i=2;i<=40;i++)
    {
        f2=f0+f1;
        f0=f1;
        f1=f2;
    }
}
```

```
cout<<"Fibonachchi sonlari ketma-ketligining "<<n<<"-hadi:"<<f2;
return 0;
}
```

### ***Amaliy topshiriqlar***

1. Berilgan  $n$  ta haqiqiy sonlar orasida qo'shnilaridan (o'zidan oldingi va keyingi sonlardan) katta bo'lgan sonlar miqdori topilsin.
2. Berilgan 10 ta natural sonlarning eng katta umumiy bo'luvchisi topilsin.
3. Berilgan  $n$  va  $m$  natural sonlari uchun  $s = \sum_{i=1}^n \prod_{j=5}^m (i+j)$  ifodaning qiymati hisoblansin.
4. Berilgan  $n$  na natural sonlardan iborat ketma-ketlikning tartib nomerlari Fibonachchi sonlari bo'lgan hadlarining yig'indisi hisoblansin.
5. Quyidagi satr chop etilsin: 1-a A , 2-b B, 3-c C, ..., 26-z Z.
6. 0 dan 15 gacha bo'lgan sonlar ikkilik ko'rinishida chop qilinsin.
7. Berilgan natural  $n$  va  $m$  uchun  $s = \prod_{i=3}^n \sum_{j=2}^m (2*i + j*j)$  ifodaning qiymati hisoblansin.
8. Berilgan natural  $n$ ,  $m$  va haqiqiy  $a$  sonlari uchun  $s = \sum_{i=4}^n \sum_{j=2}^m (a + j)$  ifodaning qiymati hisoblansin.
9. 0 bilan tugaydigan sonlar ketma-ketligi berilgan (0 soni ketma-ketlikka kirmaydi). Ketma-ketlikdagi barcha musbat sonlar yig'indisi topilsin.
10. Natural  $n$  soni va  $n$  ta haqiqiy sonlardan iborat ketma-ketlik berilgan. Toq o'rinda turgan sonlar maximumi va juft o'rindagilarning minimumi topilsin.
11. Berilgan  $n$  natural sondagi turli raqamlar miqdori aniqlansin.
12. Natural  $n$  soni va  $n$  ta sonlar juftligi berilgan -  $x_1, y_1, x_2, y_2, \dots, x_n, y_n$ . Har bir sonlar juftligi tekislikdagi nuqta koordinatasi deb hisoblab, berilgan nuqtalarni o'z ichiga oluvchi, markazi koordinata boshida bo'lgan eng kichik aylananing radiusi topilsin.
13. Raqamlari yig'indisi  $n$  soniga teng ( $1 \leq n \leq 27$ ) bo'lgan uch xonali natural sonlar soni -  $k$  topilsin. Butun sonlarni bo'lish amallaridan (/ , %) foydalanilmasin.
14. O'nlik yozuvida bir xil raqamlari bo'lmagan uch xonali sonlar o'sish tartibida chop etilsin (butun sonlarni bo'lish amalidan foydalanilmasin).
15. Taqribiy usulda  $\int_0^{3.14} \ln(2 + \sin(x)) dx$  integral to'g'ri to'rtburchaklar formulasidan foydalangan holda hisoblansin:  

$$\int_a^b f(x) dx \approx h[f(x_1) + \dots + f(x_n)],$$

bu yerda  $h = \frac{b-a}{n}$ ,  $x_i = ih - h/2$ ,  $i=1..n$ ,  $n=100$ .
16. Butun  $n$  ( $n > 1$ ) soni va  $n$  ta haqiqiy sonlardan iborat ketma-ketlik berilgan.

Ketma-ketlik o‘tuvchi yoki yo‘qligi aniqlansin.

17. Butun  $n$  ( $n > 1$ ) soni va  $n$  ta haqiqiy sonlardan iborat ketma-ketlik berilgan. Ketma-ketlikning manfiy elementlari orasidan eng kattasi topilsin.

18. Trapetsiya formulasidan -

$$\int_a^b f(x) dx \approx I_n = h \left[ \frac{f(a)}{2} + f(a+h) + f(a+2h) + \dots + f(b-h) + \frac{f(b)}{2} \right], \quad h = \frac{b-a}{n}$$

foydalanib,  $\int_c^d \cos x dx$  integrali  $\varepsilon$ -aniqlikda hisoblansin. Bu yerda  $c$ ,  $d$  va  $\varepsilon$

( $s < d, \varepsilon > 0$ ) berilgan sonlar. Zaruriy aniqlikka erishish uchun Runge qoidasidan foydalanilsin: agar  $I_n$  integralning taqribiy qiymati  $n = n_0, 2n_0, 4n_0, 8n_0$  va hokazolarda hisoblangan bo‘lib (bu yerda  $n_0$  - oraliqni boshlang‘ich bo‘lishlar soni, masalan  $n_0 = 10$ ), ular uchun  $|I_{2n} - I_n|/3 < \varepsilon$  shart bajarilganda,  $I_{2n}$  - integralning natijaviy qiymati sifatida olinsin.

19. Berilgan 80 ta haqiqiy sonlar orasidan biror butun songa eng yaqinining tartib nomeri topilsin.

20. Butun  $n$  ( $n > 1$ ) soni va  $n$  ta butun sonlardan iborat ketma-ketlik berilgan. Ulardan nechitasi eng katta qiymat qabul qilishi aniqlansin.

21. Berilgan  $n$  ta ( $n > 0$ ) haqiqiy  $x_0, x_1, \dots, x_{n-1}$  sonlardan foydalanib, quyidagi

kattaliklar hisoblansin:  $M = \frac{\sum x_i}{n}, \quad D = \sqrt{\frac{\sum (x_i - M)^2}{n-1}}.$

22. Berilgan  $x_i (i = \overline{1, 55})$  sonlari uchun  $x_0(x_1 + x_2)(x_3 + x_4 + x_5) \dots (x_{45} + x_{46} + \dots + x_{55})$  ifodaning qiymati hisoblansin.

23. 100 ta butun sonlardan iborat ketma-ketlik berilgan. Shu ketma-ketlikdagi faqat nollardan iborat eng katta ketma-ketlik ostisining uzunligi topilsin.

24. Berilgan,  $a < b$  shartni qanoatlantiruvchi  $a_1, b_1, a_2, b_2, \dots, a_n, b_n$  ( $n > 2$ ) sonlarni bir chiziqda yotuvchi kesmalarning chap ( $a$ ) va o‘ng ( $b$ ) chegaralari deb qarab, barcha kesmalar kesishmasidan hosil bo‘lgan kesma uchlari -  $(a_k, b_k)$  topilsin, agar bunday kesma mavjud bo‘lmasa, bu haqda xabar berilsin.

25. Berilgan natural sonning mukammalligi, ya'ni o‘zidan boshqa musbat bo‘luvchilarining yig‘indisi shu sonning o‘ziga tengligi aniqlansin. (misol uchun, 6- mukammal, chunki  $6 = 1 + 2 + 3$ ).

26.  $[2, n]$  ( $n > 2$ ) oraligidagi barcha tub sonlar chop etilsin.

27. Berilgan natural sonning barcha tub bo‘luvchilari chop etilsin.

28. Hadlar soni ikkitadan kam bo‘lmagan nol bilan tugaydigan natural sonlar ketma-ketligi berilgan. Tartib nomerlari tub sonlar bo‘lgan hadlar yig‘indisi hisoblansin.

29. Berilgan natural sonning raqamlarini teskari tartibda yozishdan hosil bo‘ladigan son hosil qilinsin.

30. Butun  $n$  ( $n > 0$ ) soni va  $n$  ta haqiqiy sonlar ketma-ketligi berilgan. Ketma-ketlik monotonlikka tekshirilsin.
31. O'nlik sanoq sistemasida berilgan son o'n oltilik sanoq sistemasiga o'tkazilsin.
32. Natural  $n$  soni va o'zaro teng bo'lmagan  $n$  ta butun sonlar ketma-ketligi berilgan. Bu ketma - ketlikning eng katta elementidan kichik bo'lgan sonlarning eng kattasi topilsin.
33. Berilgan ixtiyoriy  $n$  ta sonlar ichidan to'la kvadratlar miqdori aniqlansin.
34. Haqiqiy turdagi  $n$  ta sonlar juftligi berilgan.  $x_1, y_1, x_2, y_2, \dots, x_n, y_n$ . Har bir sonlar juftini tekislikdagi nuqta koordinatalari deb qarab, berilgan nuqtalar ichidan o'zaro maksimal masofadagi nuqtalar koordinatasini chop etilsin.
35. Musbat  $a$  haqiqiy soni va  $s = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n} + \dots$  qonuniyat berilgan.  $s > a$  shartni qanoatlantiruvchi birinchi  $n$  soni chop etilsin.
36. Berilgan musbat  $a$  haqiqiy sonning raqamlari yig'indisi topilsin.
37. Berilgan  $n$  ( $n > 0$ ) ta musbat sonning EKUBi topilsin.
38. Berilgan  $n$  ( $n > 0$ ) ta musbat sonning EKUKi topilsin.

## 11. while, do-while takrorlash operatorlari

### *Namunaviy masala*

Berilgan haqiqiy  $x$  va  $\varepsilon > 0$  uchun standart funksiyalardan foydalanmagan holda (fabs bundan mustasno) quyidagi ifoda hisoblansin.

$$s = e^x = 1 + \frac{x}{1} + \frac{x^2}{2} + \dots + \frac{x^n}{n!} + \dots$$

### *Yechish usuli*

Ifoda  $s = s + \frac{a}{b}$  ko'rinishdagi yig'indi masalasidir. Yig'indini hisoblash

esa har qadamda  $s$  yig'indiga  $\frac{a}{b}$  qiymatini qo'shishni amalga oshiradigan takrorlash jarayoni vositasida bajariladi. Bu yerda  $a$  va  $b$  parametrlarni hisoblash qadamiga bog'liqligini topish kerak bo'ladi. Jarayon boshlanishida ( $i=0, s=1, a=1, b=1$ ) qiymatlarni qabul qilsin. Takrorlashning  $i$ -qadamida ( $i > 1$ )  $a$  ning qiymati ( $i-1$ ) chi qadamdagi  $a$  ni  $x$  ga ko'paytirishdan, maxraj esa oldingi  $b$  ga  $i$  ni kopaytirishdan hosil bo'ladi. Yig'indini hisoblash jarayoni navbatdagi qo'shiladigan hadning absolyut qiymati  $\varepsilon$  sonidan kichik bo'lguncha davom etadi ( $\left| \frac{a}{b} \right| < \varepsilon$ ).

### *Programma matni*

```
int main()
{
    float eps, x, a=1.0, b=1.0, s=0.0;
```

```

int i=0;
cout<<" eps="; cin>>eps;
cout<<"\n x="; cin>>x;
do
{
    s+=a/b;
    a*=x;
    b*=++i;
}
while(abs(a/b)>=eps);
cout<<"exp(x)="<<s;
return 0;
}

```

### *Amaliy topshiriqlar*

- Berilgan, 7 soʻmdan katta boʻlgan har qanday tiyinsiz pul miqdorini 3 va 5 soʻmliklar yigʻindisi bilan qaytimsiz toʻlash mumkinligi isbotlansin, yaʼni berilgan  $n > 7$  uchun  $3a + 5b = n$  shartni qanoatlanti-ruvchi musbat butun  $a$  va  $b$  sonlar juftliklari topilsin.
- Berilgan natural  $n$  va  $m$  soni uchun  $t = \sum_{i=2}^n \prod_{j=3}^m (i * j)$  hisoblansin.
- Berilgan natural  $n$  va  $m$  soni uchun  $t = \prod_{i=3}^{n-1} \sum_{j=2}^{m-3} (2 * i + j)$  hisoblansin.
- Berilgan  $n$  ta sonning maksimumidan farqli sonlar ichida eng kattasi topilsin.
- Berilgan 10 ta natural sonlarning eng katta umumiy boʻluvchisi topilsin.
- Berilgan natural  $n$  sonining barcha boʻluvchilari topilsin.
- Berilgan  $n$  uchun  $s$  ning qiymati hisoblansin:
  - $s = 1! - 2! + 3! - 4! + \dots + (-1)^{n+1} n!$ ;
  - $s = -2! + 4! + \dots + (-1)^n (2n)!$ .
- Ketma-ketlik quyidagi qonuniyat bilan berilgan:
 
$$y_0 = 0; \quad y_k = \frac{y_{k-1} + 1}{y_{k-1} + 2}, \quad k = 1, 2, \dots$$

Berilgan  $\varepsilon > 0$  uchun  $y_n - y_{n-1} < \varepsilon$  shartni qanoatlantiruvchi birinchi  $y_n$  had topilsin.
- Berilgan natural  $n$  soni raqamlarining yigʻindisi hisoblansin.
- Standart funksiyalardan foydalanmagan holda ( $\text{fabs}()$ -bundan mustasno) berilgan  $\varepsilon > 0$  aniqlikda  $y$  qiymati hisoblansin. Yigʻindini hisoblashda navbatdagi qoʻshiluvchi had moduli boʻyicha  $\varepsilon$  dan kichik boʻlgan holda hisoblash jarayoni toʻxtatilsin.
  - $y = \sinh x = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + \frac{x^{2n+1}}{(2n+1)!} + \dots$ ;
  - $y = \cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots$ ;
  - $y = \ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots + (-1)^{n+1} \frac{x^n}{n} + \dots, (|x| < 1)$ ;
  - $y = \arctg x = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots, (|x| < 1)$ .

11. Bir-biridan farqli, uchtdan kam bo'lmagan natural sonlar ketma-ketligi berilgan bo'lib, u 0 bilan tugallanadi. Shu sonlar ichidan uchta eng kattasi topilsin.
  12. Nol bilan tugaydigan, noldan farqli butun sonlar ketma-ketligida ishora o'zgarishlar soni aniqlansin. (Masalan, 1,-34,8,4,-5,0 ketma-ketlikda ishora 3 marta o'zgaradi).
  13. Hadlar soni ikkitadan kam bo'lmagan nol bilan tugaydigan natural sonlar ketma-ketligi berilgan. Tartib nomerlari tub son bo'lgan hadlarning yig'indisi aniqlansin.
  14. Berilgan natural son raqamlarini teskari tartibda yozishdan hosil bo'ladigan son aniqlansin.
  15. Quyida keltirilgan ketma-ketliklarning k-raqami chop etilsin:
    - a) 12345678910111213...-ketma-ket yozilgan natural sonlar;
    - b) 149162536... - natural sonlar kvadratlari;
    - d) 1123581321...- Fibonachchi sonlari.
  16. O'nlik sanoq sistemasida natural  $p$  soni berilgan bo'lib uning  $q(2 \leq q \leq 16)$  sanoq sistemasidagi ko'rinishi hosil qilinsin.
  17. O'nlik kasr soni  $z$  uchun uning  $q(2 \leq q \leq 16)$  sanoq sistemasidagi verguldan keyingi to'rt xona aniqlikdagi ko'rinishi hosil qilinsin.
  18. O'nlik sanoq sistemasida butun  $m$  soni berilgan bo'lib, uning ikkilik sanoq sistemasidagi ko'rinishidagi sonda 0 o'rniga 1 va 1 o'rniga 0 almashtirishdan hosil bo'lgan sonning o'nlik sanoq sistemasidagi ko'rinishi aniqlansin.
  19. O'nlik sanoq sistemasidagi butun  $p$  sonining o'n oltilik sanoq sistemasidagi ko'rinishida 'E' raqami bor yoki yo'qligi aniqlansin.
  20. O'nlik sanoq sistemasidagi  $p(p < 1)$  kasr sonining oltilik sanoq sistemasiga o'tkazilganda '4' raqami necha marta uchrashi aniqlansin.
- 
21. O'nlik sanoq sistemasidagi butun  $a$  va  $b$  sonlarning uchlik sanoq sistemasidagi ko'rinishida birinchi raqamlari mos tushish yoki tushmasligi aniqlansin.
  22. Butun  $m$  va  $n$  sonlar berilgan. Umumiy bo'luvchiga ega bo'lmagan  $\frac{p}{q} = \frac{m}{n}$  shartni qanoatlantiruvchi  $p$  va  $q$  butun sonlar topilsin.
  23. Musbat haqiqiy  $a, x$  va  $\varepsilon$  sonlar berilgan.  $y_1, y_2, \dots$ -ketma-ketlik quyidagi qonuniyat bo'yicha hosil qilingan:

$$y_0 = a, y_i = \frac{1}{2} \left( y_{i-1} + \frac{x}{y_{i-1}} \right), i = 1, 2, \dots$$

Ketma-ketlikning  $|y_n^2 - y_{n-1}^2| < \varepsilon$  shartni qanoatlantiruvchi birinchi  $y_n$  hadi topilsin.

24. Berilgan

$$x_0 = 1; x_k = \frac{2 - x_{k-1}^2}{5}, k = 1, 2, \dots$$

uchun  $|x_n - x_{n-1}| < 10^{-5}$  o'rinli bo'lgan birinchi  $x_n$  hadi topilsin.

25. Haqiqiy  $x, a$  va  $\varepsilon (\varepsilon > 0, |x| < 1)$  sonlar berilgan.

$$1 + \sum_{k=1}^{\infty} \frac{a(a-1) \dots (a-k+1)}{k!} x^k$$

ifodaning qiymati  $\varepsilon$  aniqlikda hisoblansin.

26. Haqiqiy  $\varepsilon (\varepsilon > 0)$  soni va  $a, a_2, \dots$  ketma-ketlik quyidagi qonuniyat bilan berilgan:

$$a_n = \left( 1 - \frac{1}{2} \right) \left( 1 - \frac{1}{3} \right) \dots \left( 1 - \frac{1}{n+1} \right).$$

Ketma-ketlikning  $|a_n - a_{n-1}| < \varepsilon$  shartni qanoatlantiruvchi birinchi  $a_n (n \geq 2)$  hadi topilsin.

27. Haqiqiy  $a, b, \varepsilon (a > b > 0, \varepsilon > 0)$  sonlar va  $x_1, x_2, \dots, y_1, y_2, \dots$  ketma-ketliklar quyidagi qonuniyat bilan berilgan:

$$x_1 = a, y_1 = b, x_k = \frac{1}{2}(x_{k-1} + y_{k-1}), y_k = \sqrt{x_{k-1} y_{k-1}}.$$

Ketma-ketliklarning  $|x_n - y_n| < \varepsilon$  shartni qanoatlantiruvchi birinchi  $x_n$  hadi topilsin.

28. Ikkita raqobat qiluvchi  $x$  va  $y$  turlarning  $n$  - yildagi  $x_n, y_n$  - populyatsiyasilar o'lchamlariga o'zaro ta'siri quyidagi sistema bilan tavsiflanadi:

$$\begin{aligned} x_{n+1} &= 2x_n - y_n, \\ y_{n+1} &= -x_n + 2y_n. \end{aligned}$$

Faraz qilaylik  $x_0 = a$  va  $y_0 = b (a \neq b)$ . Bu yerda  $a$  va  $b$  berilgan sonlar.

Biror turning to'liq yo'qolib ketguncha har bir yildagi ikkita turning miqdorlari chop etilsin.

## 12. Funksiyalar

### Namunaviy masala

Berilgan butun sonning raqamlari yig'indisi hisoblansin.

### Yechish usuli



Berilgan butun sonning raqamlari yig'indisini hisoblash butun turdagi qiymat qaytaruvchi funksiya ko'rinishida amalga oshiriladi (`int Raqamlar_Summasi()`). Funksiya bitta butun turdagi  $n$  parametriga, yani raqamlar yig'indisi hisoblanishi zarur bo'lgan songa ega. Funksiya tanasida dastlab  $n$  soni manfiylikka tekshiriladi va u manfiy bo'lsa  $n$  ning qiymati qarama-qarshi songa aylantiriladi. Shundan keyin, toki  $n$  nolga teng bo'lmaguncha  $n$  sonini 10 ga bo'lishdagi qoldig'i  $s$  o'zgaruvchiga yig'iladi,  $n$  o'zgaruvchining yangi qiymati  $n/10$  bo'linmaning butun qismi bo'ladi. Funksiya natija sifatida  $s$  qiymatni qaytaradi.

### ***Programma matni***

```
#include<iostream.h>
int Raqamlar_Summasi(int);
int main()
{
    int b;
    cout<<"\nButun sonni kiritig:";
    cin>>b;
    cout<<"\n"<<Raqamlar_Summasi(b);
    return 0;
}
int Raqamlar_Summasi(int n)
{
    int s=0;
    if(n<0)n=-n;
    while(n>0)
    {
        s+=n%10;
        n/=10;
    }
    return s;
}
```

### ***Amaliy topshirqilar***

1. Ixtiyoriy  $n$  ta sonning yig'indisini hisoblash funksiyasi tuzilsin.
2. Ixtiyoriy  $n$  ta sonning maksimumini topish funksiyasi tuzilsin.
3. Ikkita uchburchak uchlarining koordinatalari bilan berilgan. Bu uchburchaklardan qaysi birining yuzasi katta ekanligini aniqlovchi funksiya tuzilsin.
4.  $n$  natural soni va  $n$  ta sonlardan iborat ketma-ketlik berilgan. Ketma-ketlikdagi ishora almashinishlar sonini aniqlash funksiyasi tuzilsin.
5. Berilgan  $n$  va  $m$  ( $n \geq m$ ) natural sonlar uchun Nyuton binomi koeffisientlarini

hisoblash funksiyasi tuzilsin:  $C_n^m = \frac{n!}{(n-m)!m!}$ .

6. Kompleks  $a_1 + ib_1$  va  $a_2 + ib_2$  sonlar ustida arifmetik amallar ('+', '-', '\*', '/') bajaruvchi funksiya tuzilsin, bu yerda  $a_1, b_1, a_2, b_2$  - berilgan butun sonlar.
7. Tekislikda  $a, b, c$  va  $d$  kesmalar berilgan. Ularning har bir uchtasidan uchburchak qurish mumkinligini aniqlaydigan mantiqiy Uchburchak() funksiya tuzilsin. Agar Uchburchak() funksiyasi birorta uchlik uchun true qiymat qaytarsa, shu uchlikdan hosil bo'lgan uchburchakning yuzasini hisoblovchi haqiqiy turdagi Uchburchak\_Yuzasi() funksiya tuzilsin.
8. Butun  $a, b, p, q$  ( $b \neq 0$ ) parametriga ega qisqart( $a, b, p, q$ ) funksiya  $a/b$  kasrni qisqarmaydigan  $p/q$  ko'rinishiga olib kelsin va undan  $1+1/2+1/3+\dots+1/20$  kasrni qisqarmaydigan  $c/d$  ko'rinishiga keltirishda foydalanilsin.
9. Uchburchak  $a, b$  va  $c$  tomonlari bilan berilgan. Uchburchakning medianalaridan iborat uchburchakning medianalari topilsin. (Izoh: uchburchakning  $a$  tomoniga o'tkazilgan mediananing uzunligi  $0.5\sqrt{2b^2+2c^2-a^2}$  ga teng).
10. Haqiqiy  $c, d$  ( $c < d$ ) sonlari berilgan. Quyidagi integral hisoblansin:

$$\int_c^d \arctan^2 x dx + \int_0^\pi \sin^{10} x dx.$$

Birinchi integral  $n=20$  da, ikkinchisi  $n=100$  da trapetsiya formulasidan foydalanib hisoblansin:

$$\int_a^b f(x) dx \approx h \cdot \left[ f(a)/2 + \sum_{i=1}^{n-1} f(a+ih) + f(b)/2 \right]$$

bu yerda  $h=(b-a)/n$ .

11. Har biri  $n$  ta haqiqiy elementlardan tashkil topgan  $x, y$  va  $z$  vektorlar uchun quyidagi ifoda hisoblansin ( $n$  – berilgan o'zgarmas son).

$$w = \begin{cases} \prod_i (\sin^2 x_i) + 2, & \text{agar } \prod_i (1 - y_i^2) > 0.5 \\ \prod_i (1 - z_i^2), & \text{aks holda.} \end{cases}$$

12. Berilgan  $1/(1+x^2) = x, 3e^x + x = 0$  va  $x \cdot \ln(1+x) = 0.5$  tenglamalarning yechimlari  $\varepsilon > 0$  aniqlikda topilib, o'sish tartibida chop etilsin.
13. Uchburchak uchlari va shu uchburchak ichidagi biror nuqta koordinatasi bilan berilgan. Berilgan nuqtadan uchburchak tomonlarigacha bo'lgan eng yaqin masofa topilsin.
14. Tekislikda uchta to'g'ri chiziqli  $a_k x + b_k y = c_k$  ( $k=1,2,3$ ) tenglamalari bilan berilgan. Agar bu to'g'ri chiziqlar juft-jufti bilan kesishib, uchburchak hosil qilsa, shu uchburchak yuzasi hisoblansin.
15.  $n$  dan  $2n$  ( $n > 2$ ) gacha bo'lgan sonlar ichidan "egizaklar" jufti chop etilsin. (Ikkita tub sonlar "egizak" deyiladi, agarda ular bir-biri bilan 2 ga farq qilsa, masalan: 41 va 43 sonlari).
16. Berilgan  $n$  va  $k$  uchun  $n$  sonidan boshlanuvchi  $k$  ta "tug'ma sonlar" ketma-

ketligi hosil qilinsin. Ketma-ketlikning hadi oldingi hadga uning raqamlari yig'indisini qo'shish orqali hosil bo'ladi. Masalan, ketma-ketlik hadi 13 bo'lsa undan keyingi son  $13+(1+3)=17$  bo'ladi va hokazo.

17. Berilgan natural sondan katta bo'lmagan barcha “do'st” sonlar juftligi chop etilsin (Ikkita natural son “do'st” deyiladi, agarda ularning har biri ikkinchisining bo'luvchilari (o'zidan tashqari) yig'indisiga teng bo'lsa, masalan: 220 va 284 sonlari).

18. Berilgan  $a > 0$  haqiqiy son uchun

$$\frac{\sqrt[3]{a} - \sqrt[6]{a^2 + 1}}{1 + \sqrt[7]{3 + a}}$$

kattalik hisoblansin. Bu yerda ildizni quyidagi iteratsiya formulasidan foydalanib,  $\varepsilon$  ( $\varepsilon=0.0001$ ) aniqlikda hisoblang.

$$y_0 = 1; y_{n+1} = y_n + (x/y_n^{k-1} - y_n)/k \quad (n=0,1,2,...).$$

Yechim sifatida  $|y_{n+1} - y_n| < \varepsilon$  shartni qanoatlantiruvchi  $y_{n+1}$  olinsin.

19. Berilgan haqiqiy  $\varepsilon > 0$  va  $t$  sonlari uchun

$$\sqrt[4]{1 - \frac{\cos^4 t}{4}} + \sqrt[5]{1 + \frac{\arctg t}{2}} \cdot \sqrt[9]{\frac{1}{3 + t^2}}$$

ifoda  $\varepsilon$  aniqlikda hisoblansin. Ildizlarni hisoblashda quyidagi Teylor qatoridan foydalanilsin:

$$(1+x)^a = 1 + \frac{a}{1}x + \frac{a(a-1)}{2!}x^2 + \frac{a(a-1)(a-2)}{3!}x^3 + \dots, (|x| \leq 1, a > 0).$$

20. 1, 2, ..., k nuqtalarda (bu yerda k - 2 dan 70 gacha bo'lgan butun son) quyidagi funksiyalarning grafiklari chop etilsin:

- a)  $\varphi(n)$  - 1 dan n gacha n soni bilan o'zaro tub bo'lgan butun sonlar miqdori;  
b)  $\tau(n)$  - n sonining musbat bo'luvchilari soni;  
d)  $\pi(n)$  - n sonidan oshmaydigan tub sonlar soni.

Funksiya grafigini chizishda koordinata sistemasida X o'qi pastga, Y o'qi chap tomonga yo'nalgan deb hisoblanadi. Ekrandagi belgi o'rni o'q birligi qilib olinadi. Har bir x nuqta uchun ( $x=1,2,...,k$ ) funksiya qiymati - y hisoblanadi va ekrandagi ayni satrga y-1 ta probel, undan keyin “” belgisi chop qilinadi va keyingi qatorga o'tiladi (x o'zgaruvchisining keyingi qiymati uchun).

21.  $A(x_1, y_1), B(x_2, y_2)$  va  $C(x_3, y_3)$  nuqtalar berilgan. Masalalar yechilsin:

- a) shu nuqtalardan uchburchak hosil bo'lsa true, aks holda false qiymat qaytaruvchi funksiya tuzilsin;  
b) agar ABC uchburchak mavjud bo'lsa, bu uchburchak turini aniqlovchi funksiya (teng tomonli, teng yonli, to'g'ri burchakli) tuzilsin.

22. Uchlari  $A(x_1, y_1), B(x_2, y_2), C(x_3, y_3)$  va  $D(x_4, y_4)$  nuqtalarda bo'lgan to'rtburchakni yasash mumkinmi? Agar mumkin bo'lsa uning turi aniqlansin: romb, parallelogram, trapetsia, to'g'ri burchakli yoki ixtiyoriy to'rt-burchak.

### 13. Vektorlar

#### *Namunaviy masala*

Berilgan  $n$ -darajali  $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$  va  $m$ -darajali  $b_m x^m + b_{m-1} x^{m-1} + \dots + b_1 x + b_0$  ko'phadlar ko'paytmasidan hosil bo'lgan ko'phadning koeffitsiyentlari topilsin.

#### *Yechish usuli*

$n$ - va  $m$ - darajali ko'phadlarni ko'paytirish natijasida  $n+m$  darajali ko'phad hosil bo'ladi:  $c_{n+m} x^{n+m} + c_{n+m-1} x^{n+m-1} + \dots + c_1 x + c_0$ . Asosi bir xil darajali sonlarni ko'paytirganda ularning darajalari qo'shilishi xossasidan foydalangan holda mos koeffitsientlar topiladi:  $c_{i+j} = \sum a_i * b_j$ . Har bir ko'phadning ozod hadini inobatga olgan holda ular uchun ajratiladigan massivlar o'lchami mos ko'phad darajasidan bittaga ortiq bo'lishi kerak.

#### *Programma matni*

```
int main()
{
    const int n=2+1,m=4+1;
    int a[n],b[m],c[n+m-1];
    int i,j;
    for(i=0;i<n;i++) cin>>a[i];
    for(j=0;j<m;j++) cin>>b[j];
    for(i=0;i<n+m-1;i++) c[i]=0;
    for(i=0;i<n;i++)
    for(j=0;j<m;j++)
        c[i+j]+=a[i]*b[j];
    for(i=n+m-2;i>=0;i--)
    {
        if(c[i]==0) continue;
        if(c[i]>0&&i!=(n+m-2)) cout<<" ";
        if(c[i]!=1) cout<<c[i];
        if(i>0) cout<<"x";
        if(i>1) cout<<"^"<<i;
    }
    return 0;
}
```

Yuqoridagi programmada 2 va 4- darajali ko'phadlarni ko'paytirishdan hosil bo'lgan ko'phad koeffitsientlarini hisoblash ko'rsatilgan. Programma ishga tushirilganda  $x^2 + 2x + 3$  va  $5x^4 + 4x^3 + 3x^2 + 2x + 1$  ko'phadlar koeffitsientlari kiritilsa, natija sifatida quyidagi ko'phad chop etiladi:

$5x^6 + 14x^5 + 26x^4 + 20x^3 + 14x^2 + 8x + 3$

#### *Amaliy topshiriqlar*

2. Bo'sh bo'lmagan, raqamlardan iborat va nuqta bilan tugaydigan matn berilgan. Matndagi eng ko'p uchraydigan raqam chop qilinsin (agar bunday raqamlar bir nechta bo'lsa, ulardan ixtiyoriy bittasi chop qilinsin).
3. `const int n=100;`  
`char s[n];`  
 Berilgan  $s$  vektor kattalikning indeksleri quyidagilarga teng bo'lgan elementlari chop qilinsin:  
 a) ikkinchi darajalari: (1,2,4,8,16,...);  
 b) to'liq kvadratlar: (1,4,9,16,25,...);  
 d) fibonachchi sonlari: (1,2,3,5,8,13,...).
4. `const int k=5,m=6,n=11; //n=k+m`  
`float x[k],y[m],z[n];`  
 Berilgan  $x$  va  $y$  vektorlarning har birida elementlar kamaymaydigan tartibda joylashgan. Bu ikki vektorni birlashtirishdan hosil bo'ladigan  $z$  vektorning elementlari ham kamaymaydigan tartibda joylashuvchi programma tuzilsin.
5. `char m[9],p[2];float x;`  
 Berilgan  $m$  va  $p$  vektorlar raqamlardan iborat,  $x$  o'zgaruvchiga  $0.m_1m_2...m_9 \cdot 10^{p_1p_2}$  ko'rinishidagi haqiqiy son o'zlashtirilsin.
6. `enum Oy{yan,fev,mart,apr,may,iyun,iyul,avg,sen,okt,noy,dek};`  
`float t[365];`  
`Oy oy;`  
 Kabisa bo'lmagan biror yilning har bir kuni haroratini bildiruvchi  $t$  vektor bo'yicha o'rtacha oylik harorati eng katta bo'lgan oyning nomi  $m$  aniqlansin.
7. `int x[50];`  
`bool t;`  
 Berilgan  $x$  vektorning elementlari orasida quyidagi sonlar bor yoki yo'qligiga qarab  $t$  o'zgaruvchiga `true` yoki `false` qiymat berilsin:  
 a) kamida bitta Fibonachchi soni;  
 b) kamida ikkita ikkinchi darajasi ko'rinishidagi son.
8. `char suz1[10], suz2[10];`  
`bool teng;`  
 Berilgan  $suz1$  va  $suz2$  so'zlarning har birida belgilar takrorlanmaydi. So'zlar bir-biridan qatnashayotgan belgilarning joylashuv o'rnini bilan farq qilsa, `teng` o'zgaruvchisiga `true`, aks holda `false` qiymat berilsin.
9. `const int n=20, n1=21; //n1=n+1`  
`float p[n+1],q[n+1],r[n1+1];`  
`float a;`  
 $p$  vektor bilan  $p(x) = p_0x^n + p_1x^{n-1} + \dots + p_{n-1}x + p_n$  ko'phad koeffitsiyentlari berilgan. Quyidagilar hosil qilinsin:  
 a)  $(x-a)p(x)$  ko'phadning koeffitsiyentlaridan tashkil topgan  $r$  vektor;  
 b)  $p(x+a)$  ko'phadning koeffitsiyentlaridan tashkil topgan  $q$  vektor.
10. Har biri 30 ta butun sondan iborat ikkita ketma-ketlik berilgan. Birinchi ketma-ketlikning ikkinchi ketma-ketlikka kirmagan sonlari ichidagi eng

kichigi topilsin (bunaqa sonlardan kamida bittasi mavjud deb faraz qilinsin).

11. Berilgan matn 30 ta belgidan tashkil topgan. Takrorlanuvchi belgilarni o'chirishdan hosil bo'lgan matn chop qilinsin.
12. Belgilari 100 tadan ortiq bo'lmagan va nuqta bilan tugaydigan (nuqtaning o'zi matnga kirmaydi) matndagi turli belgilar soni aniqlansin.
13. Qiymati 0 dan 20 gacha bo'lgan  $k$  butun soni berilgan.  $k$ -tartibli Chebishev ko'phadi koeffitsientlari topilsin (Izoh: Chebishev ko'phadlari  $T_n(x)$  quyidagi formula bilan aniqlanadi:

$$T_0(x)=1, T_1(x)=x, T_n(x)=2xT_{n-1}(x)-T_{n-2}(x) \quad n=2,3,\dots$$

14. Haqiqiy  $a_0, a_1, \dots, a_5$  sonlari berilgan.  $(x-a_0)(x-a_1)\dots(x-a_5)$  ko'phadning koeffitsientlari topilsin.
15. Berilgan 10-darajali  $P(x)$  va 6-darajali  $Q(x)$  ko'phadning koeffitsientlari bo'yicha  $P(Q(x))$  ko'phadning koeffitsientlari topilsin
16. 10 ta tosh bo'lib, ularning og'irliklari mos ravishda  $a_1, \dots, a_{10}$  butun sonlarga teng. Berilgan  $m_1, m_2, \dots, m_{10}$  og'irliklar uchun  $c_1, \dots, c_{10}$  hosil qilinsin. Bu yerda,  $c_k - m_k$  og'irlikni hosil qilish usullari soni, yani  $a_1x_1 + \dots + a_{10}x_{10} = m_k$  tenglamaning yechimlari, bu yerda  $x_i (i=\overline{1,10})$  o'zgaruvchisi 0 yoki 1 qiymat qabul qiladi.
17. Sonlar o'qida  $n$  ( $n>1$ ) ta son  $(a_1, a_2), (a_3, a_4), \dots, (a_{2n-1}, a_{2n})$  intervallar juftligi ko'rinishda berilgan:
  - a) intervallarning umumiy nuqtalari bormi?
  - b) Intervallar birlashmasi interval hosil qilsa, shu interval uchlari ko'rsatilsin ( $L_1$  va  $L_2$  intervallar birlashmasi deb shunday  $L_3$  intervalga aytiladiki,  $L_1$  va  $L_2$  intervallarga tegishli barcha nuqtalar  $L_3$  ga ham tegishli bo'lsa).
  - d) intervallarning birlashmasini  $n$  ta kesishmaydigan intervallar ko'rinishida taqdim etish mumkin bo'lgan son ko'rsatilsin.
  - e) kamida uchta intervalga tegishli butun sonlar bor bo'lsa, shu sonlardan birortasi ko'rsatilsin.
18. Tekislikda  $n$  ta ( $n\geq 4$ ) nuqta  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  koordinatalari bilan berilgan. Ularning ichida kvadrat hosil qiluvchi nuqtalar bor yoki yo'qligi aniqlansin.
19. Markazi  $M_i (x_i, y_i)$  nuqtada bo'lgan  $r_i$  radiusli aylanalar ( $i=\overline{1, n}$ ) berilgan. Quyidagilar aniqlansin:
  - a) aylanalar ichida uchta kesishuvchi aylanalar bormi?
  - b) alohida turgan aylanalar topilsin, ya'ni boshqa aylanalar bilan umumiy nuqtalari yo'q, birorta aylana ichida joylashmagan va boshqa aylanalarni o'z ichiga olmagan aylanalar).
20.  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  koordinatalari bilan berilgan nuqtalar to'plamining medianalar soni topilsin ( $n>2$  va  $n$  juft son). To'plam medianasi deb to'plamning ikkita ixtiyoriy nuqtasini tutashtiruvchi shunday to'g'ri

- chiziqqa aytiladiki, uning ikki tomonida bir hil sondagi nuqtalar joylashadi va bu to'g'ri chiziqda hech qanday uchinchi nuqta yotmaydi.
21. Arqon tortish musobaqasida ishtirok etmoqchi bo'lgan  $n$  ta o'quvchilar ( $n$ -juft son) ikki guruhga bo'linish uchun aylana shaklida joylashib, birdan to  $k$  sonigacha sanay boshladilar. Bunda har  $k$ -o'quvchi davradan chiqib, ikkinchi guruhga qo'shiladi. Sanoq davrada guruhlarda o'quvchilar soni teng bo'lguncha davom etadi. Har bir o'quvchining tartib nomeri boshlang'ich davrada sanoq boshlangan o'quvchidan boshlanib, sanoq yo'nalishi (soat millari yo'nalishi) bo'yicha aniqlanadi. Berilgan  $n$  va  $k$  uchun har bir guruhdagi o'quvchilar tartib nomerlari aniqlansin.
  22. Natural  $n$  soni va  $n$  ta haqiqiy sonlardan iborat ketma-ketlik berilgan. Bu ketma-ketlikni shunday tartiblangki, undagi barcha manfiy qiymatli elementlar o'zaro joylashish tartiblarini saqlagan holda ketma-ketlik boshiga ko'chirilsin va ulardan keyin musbat qiymatli elementlar ham huddi shu shart asosida joylashsin (qo'shimcha massiv ishlatilmasin).
  23. Natural  $n$  soni va  $a_1, a_2, \dots, a_n$  butun sonlardan iborat ketma-ketlik berilgan. Ketma-ketlikda bir marta qatnashgan elementlar chop qilinsin.
  24. Natural  $n$  soni va  $a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n$  sonlar ketma-ketliklari berilgan. Bu ketma-ketliklar bir-biridan faqat elementlarining joylashuv tartibi bilangina farq qilish yoki yo'qligi aniqlansin.
  25. Natural  $n$  soni,  $A = \{a_1, a_2, \dots, a_n\}$  va  $B = \{b_1, b_2, \dots, b_n\}$  sonlar ketma-ketliklari berilgan.  $A$  ketma-ketlik  $[1..n]$  oraligidagi har xil butun sonlardan tashkil topgan (indekslar).  $B$  ketma-ketlikdagi elementlar  $A$  ketma-ketlikda ko'rsatilgan joylarga o'tkazilsin, ya'ni  $b_i$  element  $a_j$  indeks bo'yicha joylashsin.
  26. Juft bo'lgan  $n$  natural soni va yarmigacha qiymatlar bilan to'ldirilgan  $A = \{a_1, a_2, \dots, a_n\}$  ketma-ketlik berilgan. Ketma-ketlik qiymatli elementlarini takrorlab surish orqali to'ldirilsin (masalan,  $A = \{3, 8, \dots\}$  uchun  $A = \{3, 3, 8, 8, \dots\}$ ).
  27. Natural  $n$  soni va  $A = \{a_1, a_2, \dots, a_n\}$  ketma-ketlik berilgan. Ketma-ketlikdagi elementlar o'sish tartibida bo'lgan eng uzun qism ketma-ketlik topilsin.
  28. Fazoda  $n$  material nuqtadan iborat sistema  $x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_n, y_n, z_n$  haqiqiy sonlar ketma-ketligi ko'rinishida berilgan. Bu yerda  $x_i, y_i, z_i$  –  $i$ -chi nuqtaning koordinatalari ( $i = \overline{1, n}$ ). Berilgan  $n$  o'zgarmas uchun sistemaning og'irlik markazi koordinatasi, hamda og'irlik markazidan sistemaning barcha nuqtalarigacha bo'lagan masofalar topilsin.
  29. Berilgan  $x, y, z$  vektorlarni element turidagi obyektlar to'plami deb qarab (agarda  $k$  element  $x$  to'plamga tegishli bo'lsa  $x[k]=true$ , aks holda  $x[k]=false$  va hokazo), ushbu vektor - to'plamlar ustida quyidagilar amalga oshirilsin:
    - a) agar  $x$  to'plam  $y$  to'plamning qism to'plami bo'lsa, u holda  $t$  o'zgaruvchiga  $true$  qiymati, aks holda  $false$  berilsin;
    - b)  $z = x \cap y$  - to'plamlar kesishmasi topilsin;

- d)  $z = x \cup y$  - to'plamlarning birlashmasi topilsin;
- e)  $z = x \setminus y$  - to'plamlarning ayirmasi topilsin ( $z$  to'plamga  $x$  to'plamning u to'plam-ga kirmagan barcha elementlari kiradi).

## 14. Matritsalar

### *Namunaviy masala*

Elementlari bir-biriga teng bo'lmagan  $n$ -chi tartibli haqiqiy turdagi kvadrat matritsa berilgan. Eng katta element joylashgan satrni eng kichik element joylashgan ustunga kopaytmasi topilsin.

### *Yechish usuli*

Matritsaning eng katta va eng kichik elementlari topiladi. Buning uchun matritsaning  $(0,0)$  o'ringagi elementi maksimum (minimum) deb faraz qilinadi va  $i$ -satr va  $j$ -ustunlar bo'icha ichma-ich joylashgan takrorlash jarayoni amalga oshiriladi. Maksimum (minimum) element topish bilan birga uning joylashgan o'ri biror maxi o'zgaruvchida (minimum uchun minj o'zgaruvchida) eslab qolinadi. Keyin matritsaning maxi - satr elementlari va minj - ustun elementlariga skalyar kopaytmasi hisoblanadi.

### *Programma matni*

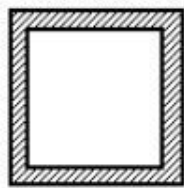
```
#include <iostream.h>
#include <conio.h>
#define n 3
typedef float Fmatritsa[n][n];
int main()
{
    Fmatritsa A;
    float max,min;
    int maxi,minj,i,j;
    //A matritsa elementlarini o'qish
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)cin>>A[i][j];
    // A[0,0]- maksimum hamda minimum element deb faraz qilinadi va
    matritsaning qolgan elementlari
    // bilan solishtiriladi;
    max=A[0][0]; maxi=0;
    min=A[0][0]; minj=0;
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
            if(max<A[i][j]){max=A[i][j]; maxi=i;}
            else if(min>A[i][j]){min=A[i][j]; minj=j;}
```



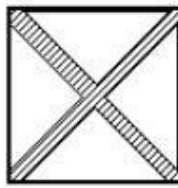
```
// Maksimum element joylashgan satr minimum joylashgan satrga
ko'paytiriladi
float S=0;
for(j=0;j<n;j++) S+=A[maxi][j]*A[j][minj];
cout<<"S="<<S;
return 0;
}
```

### ***Amaliy masalalar***

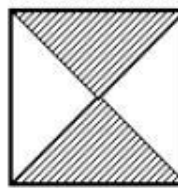
1. `const int n = 20;`  
`float B[n][n];`  
 Qo'shimcha matritsadan foydalanmagan holda **B** matritsaning transponerlangan ko'rinishi  $B^T$  hosil qilinsin.
2.  $n$  natural soni va 5-chi tartibli haqiqiy turdagi kvadrat matritsa berilgan. Bu matritsaning  $n$ -darajasi topilsin ( $A^1=A$ ,  $A^2=AA$ ,  $A^3=A^2A$  va hokazo).
3. `const int n=20;`  
`float nuqta[n][2], d;`  
 Matritsaning satr elementlarini tekislikdagi nuqtalarning koordinatalari deb qarab, shu nuqtalar orasidagi eng katta masofa topilsin.
4. `float A[9][9],s;`  
**A** matritsaning quyidagi rasmdagi bo'yalgan sohalardagi elementlari yig'indisi **s** topilsin.



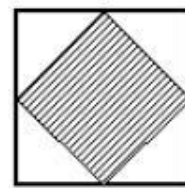
**a)**



**b)**



**d)**



**e)**

5. `enum Oy{yan,fev,mar,apr,may,iyn,iyl,avg,sen,okt,noy,dek};`  
`enum Kun{dush,sesh,chor,paysh,juma,shanba,yaksh,null};`  
`Kun Kalendar[12][31];`  
 Kalendar massivi haftaning mos kunlari bilan to'ldirilsin (mavjud bo'lmagan sana null deb ko'rsatilsin). Yil kabisa yili emas va 1-yanvar dushanba deb hisoblansin (`Kalendar[yan][0]=dush;` `Kalendar[yan][1]=sesh;... Kalendar[fev][29]=null;...` ).
6. `int A[10][10],B[9][9];`  
`int n,k; // 0 ≤ n ≤ 10 0 ≤ k ≤ 10`  
 Berilgan **A** matritsaning  $n$ -satri va  $k$ -ustunini o'chirish orqali **B** matritsa hosil qilinsin.
7. `const int n=8, m=12;`  
`int k, S[n][m];`  
**S** matritsaning "maxsus" elementlar soni -  $k$  aniqlansin.  
 Element "maxsus" deyiladi, agar u o'zi joylashgan ustundagi boshqa elementlar yig'indisidan katta va o'zi joylashgan satrda chapdagi

elementlardan katta, o'ngdagilaridan esa kichik bo'lsa.

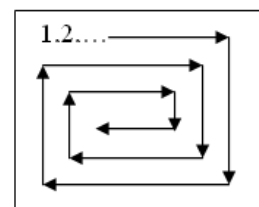
8. `int k; char C[10][15];`  
Berilgan  $C$  matritsadagi har xil belgilar soni -  $k$  aniqlansin (takrorlanuvchi belgilar bitta deb hisoblansin).
9. 5 ta satr va 7 ta ustundan iborat haqiqiy turdagi matritsa berilgan. Uning satrlari kamaymaydigan ko'rinishda tartiblansin.
10. O'lchami  $10 \times 5$  bo'lgan haqiqiy turdagi matritsa berilgan. Matritsa satrlarining eng katta elementlari o'sishi bo'yicha tartiblansin.
11. Matritsaning elementi egar nuqta deyiladi, agarda u bir vaqtning o'zida shu element joylashgan satrdagi eng kichigi va ustundagi eng kattasi bo'lsa yoki aksincha.  $10 \times 15$  o'lchamli butun turdagi matritsaning egar nuqtasi indeksi chop etilsin.
12. Elementlari bir-biriga teng bo'lmagan haqiqiy turdagi 7-tartibli kvadrat matritsa berilgan. Eng katta elementi joylashgan satrning eng kichik elementi joylashgan ustunga ko'paytmasi topilsin.
13. Butun turdagi 10-tartibli kvadrat matritsa ortonormal yoki yo'qligi aniqlansin. Matritsa ortonormal deyiladi, agar turli satrlari skalyar ko'paytmasi 0 ga teng va satrni o'z-o'ziga ko'paytmasi 1 ga teng bo'lsa.
14. Natural  $n$  soni va  $n \times n$  o'lchamli haqiqiy turdagi  $A$  matritsa berilgan. Quyidagi formula yordamida  $A$  matritsaga teskari matritsa topilsin:

$$A_k^{-1} = A_{k-1}^{-1}(2E - A * A_{k-1}^{-1}),$$

bu yerda  $A$  – berilgan matritsa;  $E$  – birlik matritsa;  $A_k^{-1}$  – teskari matritsaning  $k$ - yaqinlashishi,  $A_0^{-1} = E$ . Teskari matritsa berilgan  $\varepsilon (\varepsilon > 0)$  aniqlikda hisoblansin.

15. Natural  $n$  soni va butun turdagi  $n$ -tartibli kvadrat matritsa berilgan. Matritsa elementlari monoton ketma-ketlik hosil qiluvchi (monoton kamayuvchi yoki o'suvchi) satrlar nomerlari topilsin.
16. Butun turdagi  $n$ -tartibli kvadrat matritsaning absolyut qiymati bo'yicha eng katta elementlari toplisin. Shu elementlar joylashgan ustun va satrlarni o'chirish orqali yangi matritsa qurilsin.
17. Natural  $n$  soni va elementlari faqat 0,1, 2 va 3 sonlaridan tashkil topgan  $n$ -tartibli  $A$  matritsa berilgan. Elementlari har xil sondan iborat barcha  $a_{i,j}, a_{i,j+1}, a_{i+1,j}, a_{i+1,j+1}$  to'rtliklar miqdori topilsin.
18. Haqiqiy turdagi 9-tartibli kvadrat matritsa berilgan. Shunday matritsa hosil qiling-ki, bunda boshlangich matritsadagi mos element o'z satridagi diagonal elementdan kichik bo'lmasa bir, aks holda nol qiymat qabul qilsin.
19. `const int n=10;`  
`float A[n][n], b[n], c[n];`  
Simmetrik matritsa o'ng uchburchagi  $(n+1)*n/2$  elementlari bilan berilgan. Matritsaning 1-satrdan  $n$ -element, 2-satrdan  $n-1$ -element va oxirida  $n$ -satrdan 1-element  $b$  vektorga yuklanib,  $c=A*b$  hisoblansin.

20. `const int n=7;`  
`int A[n][n];`  
 Butun turdagi A matritsaga 1, 2, ... , 49 sonlarini spiral bo'yicha joylashtirilsin (rasmga qarang).



21. `const int n=20;`  
`int S[n][n];`  
 S matritsaning elementlarini markaz atrofida  $90^\circ$  ga soat millariga teskari yo'nalishda burish bilan qayta aniqlansin.
22. `const int n=10,m=12;`  
`int S[n][m];`  
 S matritsaning lokal minimumlari chop etilsin. Matritsaning  $s_{ij}$  elementi lokal minimum deyiladi, agar u o'z atrofidagi barcha elementlardan kichik bo'lsa.

## 15. Statik massivlar funktsiya argumenti sifatida

### *Namunaviy masala*

Berilgan butun sondagi N pul miqdori 100,50,25,10,5,3,1 qiymatdagi tangalarning eng kam miqdori bilan yig'ilsin.

### *Yechish usuli*

Berilgan N pul miqdori massivning 0-elementidan kichik bo'lmasa undan massivning 0-elementi ayriladi va u ekranga chop qilinadi. Bu jarayon toki qolgan pul miqdori massivning 0-elementidan kichik bo'lguncha davom etadi. Xuddi shu jarayon qolgan pul miqdori uchun massivning 1-elementi uchun, keyin 2-elementi uchun va shu tariqa massivning keyingi elementlari uchun takrorlanadi. Pul miqdori 0 bo'lganda jarayon to'xtaydi va ekranda zarur tangalarning tartiblangan ro'yxati chop etiladi.

### *Programma matni*

```
#include <iostream.h>
void Tangalar(unsigned int pul,unsigned int tanga[])
{
    for(int i=0;pul>=1;)
    {
        if(pul>=tanga[i]) { pul-=tanga[i]; cout<<tanga[i]<<'\\t'; }
        else i++;
    }
}
int main()
{
    unsigned int Pul,tanga[7]={100,50,25,10,5,3,1};
    cin>>Pul;
    Tangalar(Pul,tanga);
}
```

```
return 0;
}
```

Programma ishga tushirib, 239 soni kiritilsa, ekranga quyidagilar chop etiladi:

```
100 100 25 10 3 1
```

### *Amaliy topshiriqlar*

1. char Matn[72];  
char Shifr[52];  
Belgi turidagi Matn massivida lotin harflaridan tashkil topgan matn berilgan. Matn 52 ta lotin harfi uchun aniqlangan Shifr yordamida shifrlansin. Bu yerda Shifr[0] - 'A' ning, Shifr[1] - 'B' ning, ..., Shifr[51] - 'z' ning shifri bo'lgan belgi.
2. enum Hafta\_kuni{yaksh,dush,sesh,chor,paysh,juma,shanba};  
Hafta\_kuni Yil[365];  
Agar 1 yanvar- chorshanba bo'lsa (Yil[0]=chor, Yil[1]=paysh va hokazo), Yil massivining har bir elementiga, kabisa bo'lmagan yil hisobi bo'yicha i-kuniga mos keluvchi haftaning nomi (qiymati) berilsin.
3. Bo'sh bo'lmagan, raqamlardan iborat va nuqta bilan tugaydigan matn berilgan. Matndagi eng ko'p uchraydigan raqam chop qilinsin (agar bunday raqamlar bir nechta bo'lsa, ulardan ixtiyoriy bittasi chop qilinsin).
4. Har biri 30 ta butun sondan iborat ikkita ketma-ketlik berilgan. Birinchi ketma-ketlikning ikkinchi ketma-ketlikka kirmagan sonlari ichidan eng kichigi topilsin (bunaqa sonlardan kamida bittasi mavjud deb faraz qilinsin).
5. const int n=10;  
float D[n][n],S;  
  
Berilgan D matritsa uchun  $S = \sum_{k=1}^n \max_{1 \leq i, j \leq k} D_{ij}$  hisoblansin.
6. int A[15][20], b[15];  
Berilgan A matritsadan b vektor hosil qilinsin. Quyidagi ko'rsatilgan shartlar bajarilsa, b[k] elementi true, aks holda false qiymat qabul qilinsin:  
a) A matritsaning k-satri nollardan iborat;  
b) A matritsaning k-satr elementlari kamayish bo'yicha tartiblangan;  
d) A matritsaning k-satri simmetrik.
7. const int n=20;  
char Screen[n][n];  
Berilgan Screen matritsasi elementlarini markaz atrofida 90° ga soat millariga teskari yo'nalishda burish orqali qayta aniqlansin.
8. 7-tartibli elementlari bir-biriga teng bo'lmagan haqiqiy turdagi kvadrat matritsa berilgan. Eng katta element joylashgan satrning eng kichik element joylashgan ustunga ko'paytmasi topilsin.
9. Elementlari butun sonlardan iborat 10-tartibli kvadrat matritsa ortonormal yoki yo'qligi aniqlansin. Matritsa ortonormal deyiladi, agar turli satrlarni

- skalyar ko'paytmasi 0 ga teng, satrni o'ziga ko'paytmasi 1 ga teng bo'lsa.
10. Elementlari butun sonlardan iborat 9- tartibli kvadrat matritsa "*sehrli kvadrat*" ekanligi, ya'ni har bir satr va ustunlar bo'yicha elementlar yig'indilari o'zaro tengligi aniqlansin.
  11. Kichik lotin harflaridan iborat so'zlar ketma-ketligi berilgan. So'zlar bir-biridan vergul bilan ajratilgan va oxirgi so'zdan keyin nuqta qo'yilgan.  $a$  va  $b$  juftliklar ichida eng ko'p uchraydigan harflar juftligi aniqlansin (bu yerda  $a$  - ketma-ketlikdagi i-so'zning birinchi harfi,  $b$  esa oxirgi harfi).
  12. Koeffitsientlari haqiqiy turda bo'lgan  $n$  - tartibli "*uchburchak*" ko'rini-shidagi chiziqli tenglamalar sistemasi yechilsin.

$$\left. \begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\dots \\ a_{nn}x_n &= b_n \end{aligned} \right\}$$

13.  $\text{char suz1}[10], \text{suz2}[10];$   
Berilgan  $\text{suz1}$  massivdagi belgilarning o'rinlarini almashtirish orqali  $\text{suz2}$  so'zini hosil qilish mumkin yoki yo'qligi aniqlansin.
14.  $\text{const } n=4, m=5;$   
 $\text{int } A[n][m], B[n][m], C[n, m], N[n][m], D[2*n][3*m];$   
Berilgan  $A, B$  va  $C$  matritsalar orqali  $D$  matritsani quradigan  $\text{constr}(A, B, C, D)$  funksiya tuzilsin.  $D$  matritsa  $D = \begin{pmatrix} ABC \\ BNA \end{pmatrix}$  ko'rinishida aniqlanadi. Bunda  $N$  nol matritsa.
15. Uchta haqiqiy turdagi 4-tartibli kvadrat matritsalar berilgan. Ularning orasida normasi eng kichik bo'lgani chop etilsin (bunday matritsa bitta deb hisoblansin). Matritsaning normasi sifatida uning elementlari absolut qiymatlarining maksimumi olinsin.
16. Natural  $p$  soni va 4-tartibli haqiqiy turdagi  $A, B$  va  $C$  kvadrat matritsalar berilgan.  $(ABC)^p$  hosil qilinsin
17. Haqiqiy turdagi  $10 \times 20$  o'lchamli  $A, B$  va  $C$  matritsalar berilgan. Quyidagi kattalik hisoblansin:  
$$\frac{\|A\| + \|B\| + \|C\|}{\|A+B+C\|}, \text{ bu yerda } \|D\| = \max_j |D_{1,j}| + \max_j |D_{2,j}| + \dots + \max_j |D_{10,j}|.$$
18. Ikkita 10-tartibli butun turdagi kvadrat matritsa berilgan. Matritsa diagonallariga nisbatan bir marta akslantirish orqali biridan ikkinchisini hosil qilish mumkinmi?
19.  $2^{500}$  va  $1+2+3+\dots+100$  sonlarining o'nlik yozuvidagi barcha raqamlari chop etilsin. (Ko'rsatma: "*uzun*" natural sonlarni raqamlardan iborat massiv ko'rinishida ifodalab, kerakli amallar bajarilsin.)
20.  $n$  ( $n=100$ ) ta haqiqiy sonlar berilgan. Fon Neyman usuli bilan ular o'sish

tartibida joylashtirilsin: ikkita **A** va **B** massividan foydalaniladi. Berilgan sonlar **A** massiviga yoziladi; keyin yonma-yon sonlar tartiblanib ( $A_1$  va  $A_2$ ,  $A_3$  va  $A_4$  va hokazo) **B** ga yoziladi; **B** dagi ikkita yonma-yon turgan, tartiblangan juftlik olinib, ular tartiblangan to'rtlikka o'tkaziladi va yana **A** ga yoziladi; keyin **A** dan har ikkita yonma-yon to'rtlikni tartiblab, sakkizlik sifatida **B** ga yoziladi va hokazo.

21. 2 dan 20 gacha bo'lgan butun  $n$  va haqiqiy  $\varepsilon > 0$  sonlari berilgan. Quyidagi formula bilan aniqlanadigan  $n$ -tartibli  $T_n(x)$  Chebishev ko'phadining (13.12 masalaga qarang) barcha ildizlari  $\varepsilon > 0$  aniqlikda hisoblansin.  
*Izoh:* agar  $(-1, 1)$  intervalda  $x_1 < x_2 < \dots < x_k$  -  $T_k(x)$  ko'phadning ildizlari bo'lsa, u holda  $T_{k+1}(x)$  ko'phad quyidagi  $(-1, x_1), (x_1, x_2), \dots, (x_k, 1)$  intervallarda bittadan ildizga ega bo'ladi.
22. Tasodifiy sonlarni hosil qiluvchi funksiyadan foydalangan holda  $n$  ta o'zaro teng bo'lmagan butun sonlar massivini natija sifatida qaytaradigan funksiya tuzilsin.
23. Tasodifiy sonlarni hosil qiluvchi funksiyadan foydalangan holda butun turdagi  $n$ -tartibli kvadrat matritsa hosil qilinsin ( $n=10$ ). Matritsaning eng katta va eng kichik elementlarini almashtiruvchi funksiya tuzilsin.
24. Elementlari musbat bo'lgan ( $n=10$ )  $a[n][n]$  massiv berilgan. Massivning ikkinchi maksimumini qaytrardigan funksiya tuzilsin.

## 16. Dinamik massivlar funksiya argumenti sifatida

### *Namunaviy masala*

Berilgan  $n \times m$  o'lchamdagi matritsaning  $k$ -ustundagi va  $l$ -satrdagi eng katta elementlari topilsin ( $0 \leq k \leq m$   $0 \leq l \leq n$ ).

### *Yechish usuli*

O'lchami programma ishlash jarayonida kiritilishi sababli bosh funksiyada matritsa dinamik ravishda butun sonlar vektorlariga (matritsa satrlariga) ko'rsatgichlar vektori ko'rinishida aniqlanadi. Berilgan ustundagi va satrdagi maksimal elementlarni topish uchun alohida funksiyalar aniqlanadi va zarur parametrlar ularga argument sifatida uzatiladi. Shuni qayd etish kerakki,  $l$ -satr eng katta elementini topadigan funksiyani chaqirganda argument sifatida matritsaning mos satriga ko'rsatgichni uzatish yetarli,  $k$ -ustun maksimumini topishda esa funksiya argument sifatida matritsaga ko'rsatgichni uzatish kerak bo'ladi.

### *Programma matni*

```
int K_ustun_max(int n,int k,int **a)
{
    int max=a[0][k];
```

```

    for(int i=1;i<n;i++)
    if(max<=a[i][k]) max=a[i][k];
    return max;
}
int L_satr_max(int m,int *satr)
{
    int max=satr[0];
    for(int i=1;i<m;i++)
    if(max<=satr[i]) max=satr[i];
    return max;
}
int main()
{
    int n,m,k,l;
    int **a;
    cout<<"Matritsaning satrlar soni="; cin>>n;
    cout<<"Matritsaning ustunlar soni="; cin>>m;
    // matritsani hosil qilish
    a=new int *[n];
    for(int i=0; i<n; i++) a[i]=new int[m];
    // matritsa elementlari qiymatlarini kiritish
    for(int i=0; i<n; i++)
    for(int j=0; j<m; j++) cin>>a[i][j];
    // matritsani chop etish
    cout<<"Matritsa ko'rinishi:\n";
    for(int i=0; i<n; i++)
    {
        for(int j=0; j<m; j++)
            cout<<a[i][j]<<' ';
        cout<<endl;
    }
    do
    {
        cout<<"k-ustun nomerini kiriting="; cin>>k;
    }
    while(k<0 || k>m);
    do
    {
        cout<<"l-ustun nomerini kiriting="; cin>>l;
    }
    while(l<0 || l>n);
    cout<<k<<"-ustun maximal elementi ="<<K_ustun_max(n,k,a);
    cout<<"\n"<<l<<"-satr maximal elementi =";
    cout<<L_satr_max(m,a[l]);
}

```

```

for(int i=0; i<n; i++)
delete[]a[i];
delete[]a;
return 0;
}

```

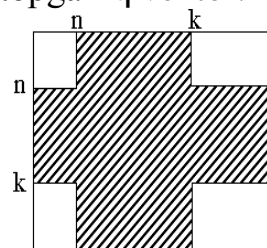
### *Amamliy topshiriqlar*

3. int \*s;  
Berilgan s vektor kattalikning indeksleri quyidagilarga teng bo'lgan elementlari chop qilinsin:  
a) ikkinchi darajalari: (1,2,4,8,16,...);  
b) to'liq kvadratlar: (1,4,9,16,25,...);  
d) Fibonachchi sonlari: (1,2,3,5,8,13,...);
4. float \*x;  
Berilgan x vektor orqali  $\vec{X}$  vektor quyidagi qoida bo'yicha hosil qilinsin ( $x_k$  hosil bo'lgan vektorning k-elementi qiymati):  
a)  $x_k = \max \{x_i\}$  bunda  $1 \leq i \leq k$ ;  
b) vektor elementlari teskari tartibda joylashtirilsin;  
d)  $x_1 = x_1, x_n = x_n, x_k = (x_{k-1} + x_k + x_{k+1})/3, k=2,3,...,n-1$ ;  
e) vektor elementlari siklik ravishda p taga chapga surilsin.
5. float \*x;  
Berilgan x vektorning barcha manfiy elementlari uning boshiga o'tkazilsin. Hosil bo'lgan vektorda manfiy va qolgan elementlarining boshlang'ich o'zaro joylashuvi saqlansin (qo'shimcha vektordan foydalanilmasin).
6. int \*x,\*y,\*z;  
Berilgan x va y vektorlarning har birida elementlar kamaymaydigan tartibda joylashgan. Bu ikki vektorni birlashtirib, shunday z vektor hosil qilinsinki, uning elementlari ham kamaymaydigan tartibda bo'lsin.
7. char \*suz1, \*suz2;  
bool teng;  
Massiv ko'rinishida berilgan suz1 va suz2 so'zlarning har birida belgilar takrorlanib kelmaydi deb hisoblagan holda, ularda qatnashayotgan belgilarning joylashuvi bilan farq qilsa, teng o'zgaruvchiga true, aks holda false qiymat berilsin.
8. float \*x;  
Berilgan x vektor almashtirish usuli orqali kamaymaydigan ko'rinishda tartiblansin ("*pufakchalar usuli*"). Bu usulning 1-qadamida qo'shni  $x_k$  va  $x_{k+1}$  ( $k=1,2,3,...,n-1$ ) elementlar ketma-ket solishtiriladi va agarda  $x_k > x_{k+1}$  bo'lsa, ularning o'rnini almashtiriladi. Shu yo'l bilan eng katta elementni vektorning oxirida joylashuviga erishiladi. Keyingi qadamda bu usul oxirgi elementdan tashqari barcha elementlarga qo'llaniladi va h.k. Jarayonning birorta qadamida elementlarni almashtirish ro'y bermasa tartiblash to'xtatilsin.



9. `float *x;`  
Berilgan  $x$  vektor orasiga qo'yish usuli orqali kamaymaydigan ko'rinishda tartiblash. Bu usulda vektorning birinchi  $k$  ta elementi kamaymaydigan ko'rinishda tartiblangan deb hisoblanadi;  $k+1$  - elementi olinadi va u birinchi  $k$  ta element orasiga shunday joylashtiriladiki, hosil bo'lgan  $k+1$  ta element tartiblangan bo'ladi; bu usul  $k$  o'zgaruvchining 0 dan  $n-2$  gacha qiymatlari uchun takrorlanadi.
- 10 Nuqta bilan tugaydigan kichik lotin iborat matn berilgan. Shu matnga faqat bir martadan kiruvchi barcha harflar alfavit tartibida chop qilinsin.
- 11 `float **A,**B,**C; // n*n o'lchamli matritsalar`  
`float *x,*y; // n o'lchamli vektor`  
Berilgan natural  $n$  uchun quyidagilar hisoblansin:  
a)  $C=A+B$ ;      b)  $y=Ax$ ;      d)  $C=AB$ .
- 12 Berilgan butun  $n$  va  $m$  boyicha  $n \times m$  o'lchamli haqiqiy turdagi dinamik matritsa berilgan. Uning satrlari kamaymaydigan ko'rinishda tartiblansin:  
a) birinchi elementlar bo'yicha;  
b) elementlar yig'indisi bo'yicha;  
d) eng katta elementlari bo'yicha.
- 13 Shaxmat musobaqasida qatnashgan  $n$  ta shaxmatchining natijalari  $T$  jadvalda berilgan ( $n > 2$ ):  
`enum Uyin_Natijasi{Y,D,M,X};`  
`Uyin_Natijasi` turidagi  $n \times n$  o'lchamli dinamik massiv (jadval) hosil qilinsin. Uning qiymatlari quyidagicha aniqlasin:  $jadval[i][j]=Y$ , agar  $i$  - ishtirokchi  $j$  - ishtirokchi ustidan g'alaba qozongan bo'lsa, (bunda  $jadval[j][i]=M$ ),  $jadval[i][j]=D$  va  $jadval[j][i]=D$ , agar  $i$  - va  $j$  - ishtirokchilar durang o'ynagan bo'lsa, hamda  $jadval[i][i]=X$ .  
Jadvalning ko'rinishi quyidagicha bo'lishi mumkin (rasmga qarang). Yutuq uchun 3 ochko, durang uchun 1 ochko, mag'lubiyat uchun 0 ochko beriladi. Ishtirokchilar nomarlari ularning to'plagan ochkolari bo'yicha o'smaydigan tartibda chop qilinsin.
- |   |   |   |
|---|---|---|
| X | Y | M |
| M | X | D |
| Y | D | X |
- 14 Berilgan  $n \times m$  o'lchamli butun turdagi dinamik ravishda yaratilgan matritsaning egar nuqtasi indeksleri chop etilsin. Matritsa elementi "*egar nuqta*" deyiladi, agarda u bir vaqtning o'zida shu element joylashgan satrdagi eng kichik va ustundagi eng katta element bo'lsa yoki aksincha.
- 15  $n$  o'lchovli chiziqli fazoda  $m$  ta vektor koordinatalari bilan berilgan. Bu vektorlar chiziqli erkli bo'ladimi? Vektorlar chiziqli erkli deyiladi, agar har qanday  $i \neq j (1 \leq i, j \leq m)$  uchun  $x_i = \alpha x_j$  sharti o'rinli bo'lmasa, bu yerda  $\alpha > 0$ .
- 16 Elementlari  $n$  ta haqiqiy sonlardan iborat  $x, y$  va  $z$  vektorlar berilgan.  $(a,a)-(b,c)$  kattalik hisoblansin, bu yerda  $a$  vektor berilgan vektorlar ichidan eng katta minimal elementga ega (bunday vektor yagona deb hisoblansin),  $b$  va  $c$  qolgan ikkita vektor,  $(p,q)-p$  va  $q$  vektorlarning skalyar ko'paytmasi.

- 17  $9 \times 4$  o'lchamli uchta butun turdagi matritsalar dinamik ravishda aniqlanib qiymatlari berilgan. Faqat nollardan iborat satrlari eng ko'p bo'lgan matritsa chop etilsin (agar bunday matritsalar bir nechta bo'lsa, barchasi chop etilsin).
- 18  $n$  ta butun elementli dinamik aniqlangan  $x$  vektorning elementlari o'sish bo'yicha tartiblangan.  $k$  o'zgaruvchiga berilgan  $p$  soniga teng bo'lgan  $x$  vektori elementining tartib nomeri berilsin, agarda bunday element bo'lmasa,  $k$  o'zgaruvchisiga  $-1$  berilsin.
- Masalani yechish uchun quyidagi ikkilik (binar) qidirish usuli qo'llanilsin:  $p$  sonni vektor o'rtasidagi yoki unga eng yaqin element bilan solishtiriladi; agar bu sonlar teng bo'lsa qidirish to'xtatiladi; agar  $p$  son elementdan kichik bo'lsa, u holda  $p$  sonini vektorning chap yarmidan izlash kerak, aks holda o'ng yarmidan; vektorning tanlangan yarmiga yuqoridagi algoritim qo'llaniladi.
- 19  $n$  ta satr va  $m$  ta ustundan iborat  $A$  va  $B$  matritsalarining maksimal elementlarini almashtiruvchi  $\text{swap}(A,B)$  funksiyasi tuzilsin. (Har bir matritsada maksimal element bitta deb hisoblansin).
- 20  $p(x) = p_0x^n + p_1x^{n-1} + \dots + p_{n-1}x + p_n$  ko'phad ko'effitsiyentlari bilan berilgan. Berilgan haqiqiy  $x$  va  $a$  sonlari uchun quyidagilar hosil qilinsin:
- $(x-a)p(x)$  ko'phadning ko'effitsiyentlaridan tashkil topgan  $r$  vektor;
  - $p(x+a)$  ko'phadning ko'effitsiyentlaridan tashkil topgan  $q$  vektor.
- 21 Butun turdagi  $N \times N$  o'lchamli dinamik ravishda aniqlangan qiymatlardan tashkil topgan  $A$  matritsaning bo'yalgan sohasidagi katta elementini topuvchi  $\text{max\_Paint}(A,n,k)$  funksiyasi tuzilsin. Bu yerda  $A$ -matritsa,  $n$  va  $k$ -indekslar bo'lib (rasmga qarang), ularning to'g'ri kiritilishi nazorat qilinsin.
- 22 Klaviaturadan natural  $n$  soni va  $n$  ta haqiqiy sonlar kiritilsin. Haqiqiy sonlarni o'qish jarayonida har bir o'qilgan son  $a$  vektorga, undagi kamaymaydigan tartibni saqlagan holda joylansin.



## 17. Satrlar

### *Namunfviy masala*

ASCIIZ satr ko'rinishida berilgan gapdagi simmetrik so'zlar chop etilsin. Gap nuqta bilan tugaydi va undagi so'zlar bir-biridan ',' (vergul) yoki '\_' (probel) bilan ajratilgan.

### *Yechish usuli*

Qo'yilgan masalani yechish uchun takrorlanuvchi ikkita amallar bajarish kerak bo'ladi:

- 1) satrdan so'zlarni (satr ostilarini) ajratib olish;
- 2) satr ostini simmetrikka tekshirish, agar natija ijobiy bo'lsa satrni chop etish.

Satrdan satr ostilarini ajratib olish uchun <string.h> kutubxonasida mavjud strtok() funksiyasidan foydalanish mumkin. strtok() prototipi

char\* strtok(char\* str, const char \* delim);

ko'rinishida bo'lib, u str satrdagi delim satr-ro'yxatda berilgan ajratuvchilar oralig'iga olingan satr qismini qiymat sifatida qaytaradi. Funksiya qaytargan satr qismini simmetriklikka tekshirish uchun satr-vektorning markaziga nisbatan teng uzoqlikda joylashgan elementlarning o'zaro bir xil ekanligi asosida tekshiradigan mantiqiy funksiyani tuzish orqali echiladi.

### ***Programma matni***

```
#include <iostream.h>
#include <string.h>
const N=80;
bool Simmetrik(const char * s)
{
    int n=strlen(s);
    for(int i=0;i<=n/2;i++)
        if(s[i]!=s[n-i-1]) return false;
    return true;
}
int main()
{
    char Gap[N];
    gets(Gap);
    char Ajratuvchi[]=" .,.";
    char* Gap_Osti;
    Gap_Osti=strtok(Gap,Ajratuvchi);
    if(Gap_Osti && Simmetrik(Gap_Osti))cout<<Gap_Osti<<"\n";
    while(Gap_Osti)
    {
        Gap_Osti=strtok(NULL,Ajratuvchi);
        if(Gap_Osti && Simmetrik(Gap_Osti))cout<<Gap_Osti<<"\n";
    }
    return 0;
}
```

### ***Amamliy topshiriqlar***

1. char t[100];  
Berilgan t satrdagi bir-biridan farqli bo'lgan kichik lotin harflari alfavit tartibida chop qilinsin.
2. char s[80];  
Berilgan s satr quyidagicha o'zgartirilsin:  
a) satrdagi barcha "bir" satrostilari " \_1\_" satrostiga almashtirilsin (bu yerda '\_' - probel);

- b) satrga kiruvchi oxirgi 'x' harfi (agarda u mavjud bo'lsa) "ks" ga almashtirilsin;
- d) satrga kiruvchi barcha "th" belgilar o'chirilsin;
- e) satrga kiruvchi har bir 'q' harfidan keyin 'u' harfi qo'shilsin.
- 3. char s1[10],s2[10],s3[11];  
O'nlik sanoq sistemasidagi butun s1, s2 sonlarning razryad bo'yicha yig'indisi s3 o'zgaruvchisida hosil qilinsin.
- 4. char bayt[8];  
Berilgan butun n ( $0 \leq n \leq 127$ ) soniga qarama-qarshi sonning (manfiy sonning) qo'shimcha koddagi ikkilik ko'rinishi bayt massivida hosil qilinsin.
- 5. char gap[80];  
Berilgan gapdagi so'zlar bir-biridan ',' yoki '\_' (probel) belgisi bilan ajratilgan va gap nuqta bilan tugaydi (gap o'zgaruvchisining qiymati).  
Quyidagilar chop qilinsin:
  - a) ketma-ketlikda faqat bir marta uchragan so'zlar;
  - b) ketma-ketlikda bir necha bor uchragan so'zlar;
  - d) ketma-ketlikdagi barcha so'zlar alfavit tartibida.
- 6. char gap[80];  
Berilgan gapdagi so'zlar bir-biridan ',' yoki '\_' (probel) belgisi bilan ajratilgan va gap nuqta bilan tugaydi (gap o'zgaruvchisining qiymati).  
Quyidagi shartlarni bajaruvchi so'zlar chop qilinsin:
  - a) birinchi harfi yana uchragan;
  - b) eng uzun;
  - d) harflari takrorlanmaydigan.
- 7. Har biri 2 tadan 10 tagacha lotin harflaridan iborat 2 dan 30 gacha bo'lgan so'zlar ketma-ketligi berilgan. So'zlar bir-biri bilan kamida bitta probel bilan ajratilgan va oxirgi so'z nuqta bilan tugaydi. Ketma-ketlikning oxirgi so'zidan farqli barcha so'zlar quyidagi qoida bo'yicha o'zgartirilsin:
  - a) so'zdagi birinchi harfi uning oxiriga o'tkazilsin;
  - b) so'zdagi oxirgi harf uning boshiga o'tkazilsin;
  - d) so'zdagi birinchi harf olib tashlansin.
- 8. Har biri 2 tadan 10 tagacha lotin harflaridan iborat 2 dan 30 gacha bo'lgan so'zlar ketma-ketligi berilgan. So'zlar bir-biri bilan kamida bitta probel bilan ajratilgan va oxirgi so'z nuqta bilan tugaydi. Ketma-ketlikning oxirgi so'zidan farqli barcha so'zlari quyidagi qoida bo'yicha o'zgartirilsin:
  - a) so'zdagi oxirgi harf takrorlansa, oxirgisidan boshqa takrorlanganlari olib tashlansin;
  - b) so'zdagi har bir harfni faqat birinchi marta qatnashgani qoldirilib, qolganlari olib tashlansin;
  - d) agar so'zning uzunligi toq songa teng bo'lsa, u holda uning o'rtasidagi harf olib tashlansin.
- 9. 1 dan 1999 gacha bo'lgan butun n sonining Rim raqamlaridagi ko'rinishi chop qilinsin.
- 10. Raqam va to'rtta arifmetik amallardan ('\*', '+', '-', '/') tashkil topgan ifoda

postfiks ko‘rinishiga o‘tkazilsin. Postfiks shaklda oldin operandlar yoziladi, keyin amallar.

Misollar:	oddiy yozuv	postfiks yozuv
	3+4	34+
	(5-4)+2	54-2+
	2*(3+4)*5	234+*5*

11. Satrda so‘zlar ketma-ketligi berilgan. So‘zlar bir-biridan kamida bitta probel bilan ajratilgan. Oxirgi so‘z nuqta bilan tugaydi. So‘zlar alfavit bo‘yicha tartiblansin.
12. `char s1[10],s2[10];`  
`char op;`  
 O‘noltilik sanoq sistemasida berilgan `s1` va `s2` butun sonlar ustida `op` (\*, +, -, / ) amali bajarilsin.
13. 20 ta so‘zdan iborat lug‘at yordamida faqat shu so‘zlardan tuzilgan sodda gap boshqa tilga tarjima qilinsin.
14. Berilgan matndagi harflarning qatnashisih chastotasi aniqlansin va bu harflar chastotalarining kamayishi bo‘yicha chop qilinsin.
15. `const int n=20,m=40;`  
`char matn[n][m];`  
 Har bir satri `m` belgigacha bo‘lgan `n` ta satrdan iborat matn berilgan. Matn `m` uzunligidagi o‘ng va chap tomondan tekislangan satrlarga o‘tkazilsin. Bunda satrda kamida ikkita so‘z bor deb hisoblanadi. Qisqa satrlar so‘zlar orasiga probellarni qo‘yish bilan to‘ldiriladi.
16. Berilgan so‘zni quyidagi qoida yordamida ikkiga bo‘lishni amalga oshiring (keyingi satrga o‘tkazish uchun):
  - 1) ketma-ket kelgan ikkita unli harfni ajratish mumkin, agar birinchisidan oldin undosh harf va ikkinchisidan keyin kamida bitta harf kelsa;
  - 2) ketma-ket kelgan ikkita undosh harfni ajratish mumkin, agar birinchisidan oldin unli harf va ikkinchisidan keyin soz bo‘lagidagi kamida bitta unli harf kelsa (‘’, - ajratish belgilari o‘zidan oldindagi harf bilan bitta deb qaraladi);
  - 3) agar 1) va 2) qoidalarni qo‘llash mumkin bo‘lmasa so‘zni shunday bo‘lish kerakki, unig birinchi qismi kamida bitta unli harfni o‘z ichiga olishi va unli bilan tugallanishi kerak, ikkinchi qism kamida bitta unli harfni o‘z ichiga olishi kerak.
17. Berilgan `n` butun soni ( $0 \leq n \leq 999999$ ) so‘zlar orqali yozilsin. Masalan, 15 soni “o‘n besh” ko‘rinishida.
18. Ikkilik sanoq sistemasidagi son satr ko‘rinishida berilgan. Shu sonning 8 va 16 lik sanoq sistemasidagi ko‘rinishi chop qilinsin.
19. `char x[10];`  
 Berilgan `x` satri identifikator yoki yo‘qligi aniqlansin.
20. `typedef char Misra[60];`  
`typedef Misra Turtlik[4];`  
`Turtlik she_r;`  
 Berilgan `she_r` turtlikda qofiya bor yoki yo‘qligi aniqlansin.
21. `const int n=60;`

char jumla1[n], jumla2[n];

Begilgan jumla1 va jumla2 satrlar bir-biridan faqat ulardagi soʻzlarning gapdagi joyla-shuv oʻrni bilan farq qilishi aniqlansin.

22. char s[60];

Berilgan satrda polindrom soʻzlar mavjud boʻlsa ular chop qilinsin, aks holda bunday soʻz yoʻqligi haqida xabar berilsin (satrdagi soʻzlar bir-biridan bitta yoki bir necha probel orqali ajratiladi).

23. const int n=30;

typedef char FIO[40];

FIO Guruh[n];

unsigned int Tug\_Yil[n];

Oʻquv guruhi (Guruh) talabalarining roʻyxati quyidagi tartibda: talabaning familiyasi, ismi va otasi ismining bosh harflari hamda tugʻilgan yili nuqta bilan ajratilgan koʻrinishda berilgan. Roʻyxat lotin alifbosidagi harflar joylashuvi boʻyicha tartiblansin. Tatriblash talaba familiyasi va initsiallari boʻyicha amalga oshirilsin. Agar bir xil familiya va initsialli talabalar uchrasa, ular roʻxatda tugʻilgan yilini (Tug\_Yil) kamayishi boʻyicha joylashtirilsin.

24. const int n=60;

char Jumla[n];

Berilgan gap(Jumla)dagi har bir soʻzning birinchi harfi bosh harfga almashtirilsin.

25. const int n=60;

char Jumla[n];

Lotin harflaridan tashkil topgan Jumla satri berilgan. Satrdagi har bir harf alfavitdagi oʻzidan keyingi harf bilan almashtirish orqali shifrlansin va qayta tiklansin. Almashtirishda harflar registri inobatga olinsin ('A' harfi oʻtkazilsin 'B' harfiga, 'a'→'b', 'B'→'C', 'z'→'a' va hokazo). Satrdagi ajratuvchilar (probel, ',', va boshqalar) oʻzgartirilmasin.

26. const int n=60;

char Jumla[n];

Lotin harflaridan tashkil topgan Jumla satri berilgan. Satr, undagi juft oʻrinda turgan harflarni satr boshiga oʻtkazish va toq oʻrindagi harflarni teskari tartibda qayta joylashtirish orqali shifrlansin va qayta tiklansin. Masalan, "Programma" satri "rgamamroP" satriga aylanadi.

27. char \*fam[ ]; char \*ism[ ]; int tel[ ];

Yuqorida keltirilgan massivlarda mos ravishda familiyalar, ismlar va telefon nomerlar berilgan. Soʻralgan ism (familiya)ga mos telefon nomerni yoki soʻralgan telefonga mos ism (familiya)ni topuvchi, agarda bunday maʼlumot mavjud boʻlmasa uning yoʻqligi haqida xabar beruvchi programma tuzilsin.

28. Butun musbat n, m sonlari va char a[n][m] massiv berilgan. a massivning har bir satridagi raqamlar miqdori aniqlansin.

29. char \*\*familiya,\*\*ism,\*\*telefon;

Yuqorida keltirilgan massivlarda mos ravishda familiyalar, ismlar va telefon nomerlar berilgan (bu massivlarning har qatorlari oʻzaro bir qiymatli boʻgʻliq).

- a) familiya massivining alfavit bo'icha tartiblanishiga mos ravishda ism va telefon massivlari ham tartiblansin;
  - b) berilgan telefon nomerining boshlangich bo'lagi bilan mos tushuvchi nomerlarga mos familiya va ismlar chop qilinsin.
30. Satr va  $n$  ( $n > 0$ ) natural soni berilgan. Satrdagi  $n$ -so'z o'chirilsin.
31. Berilgan matn teskariga o'girilsin, yani undagi har bir so'z teskari yozilsin (masalan: "Uzmu 1-kurs talabasi"  $\rightarrow$  "isabalat sruk-1 umzU").
32. Berilgan  $S$  satrda ikkita bir xil belgilar takrorlanmaydigan eng uzun satrosti chop etilsin.

## 19. Rekursiv funksialar

### *Namunaviy masala*

Tezkor tartiblash usuli yordamida  $n$  ta ( $n > 0$ ) butun sonlardan iborat  $x$  massiv kamaymaydigan ko'rinishda tartiblansin.

### *Yechish usuli*

Tezkor tartiblash usulining mohiyati shundaki,  $x$  massivning birorta elementi (masalan, o'rtadagi elementi)  $xi$  olinadi. Massiv elementlari shunday almashtirladiki, natijada massivning  $xi$  element joylashgan joydan chap tomonidagi qismida faqat undan katta bo'lmagan elementlar, o'ng tomonda undan kichik bo'lmagan elementlar joylashib qoladi, y'ani  $xi$  tartiblangan ketma-ketlikdagi o'zing yakuniy o'rnini topadi. Keyingi qadamda  $xi$  element joylashgan joydan chap va o'ng tomondagi massiv qismlariga yuqoridagi usul qo'llaniladi va hokazo.

### *Programma matni*

```
void Tezkor_Tartiblash(int *a, int i_chap, int i_ung)
{
    int chap, ung, ai;
    if (i_chap >= i_ung) return;
    chap = i_chap;
    ung = i_ung;
    ai = a[(chap+ung)/2];
    while(chap <= ung)
    {
        while(a[chap] < ai) chap++;
        while(a[ung] > ai) ung--;
        if(chap <= ung)
        {
            int vaqtincha = a[chap];
            a[chap] = a[ung];
            a[ung] = vaqtincha;
            chap++;
            ung--;
        }
    }
}
```

```

    }
}
Tezkor_Tartiblash(a, i_chap, ung);
Tezkor_Tartiblash(a, chap, i_ung);
}
int main()
{
    int *x;
    int n;
    cin>>n;
    x=new int [n];
    for (int i=0;i<n;i++) cin>>x[i];
    cout<<"Berilgan massiv"<<endl;
    for(int i=0;i<n;i++) cout<<x[i]<<"\t";
    cout<<endl;
    Tezkor_Tartiblash(x,0,n-1);
    cout<<"Tartiblangan massiv:"<<endl;
    for(int i=0;i<n;i++) cout<<x[i]<<"\t";
    return 0;
}

```

### *Amailiy masalalar*

1.  $C_n^0 = C_n^n = 1; C_n^m = C_{n-1}^m + C_{n-1}^{m-1}$  formula bo'yicha  $C_n^m$  binom koeffisientini hisoblaydigan  $C(n,m)$  rekursiv funksiya tuzilsin, bunda  $0 \leq m \leq n$ .
2. Quyidagi

$$x^n = \begin{cases} 1, & \text{agar } n=0, \\ 1/x^{|n|}, & \text{agar } n<0, \\ x \cdot x^{n-1}, & \text{aks holda.} \end{cases}$$

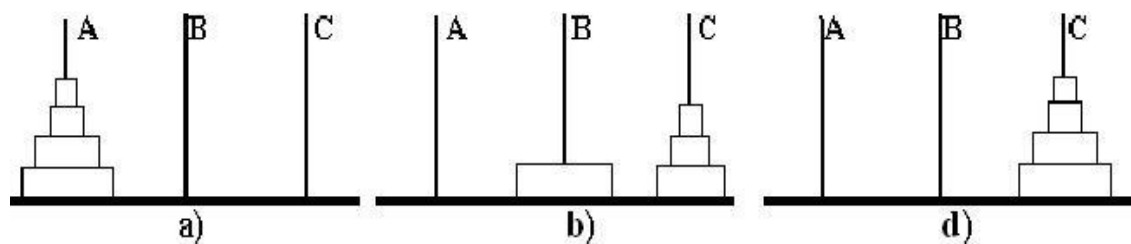
formula yordamida  $x^n$  kattalikni hisoblovchi  $\text{Daraja}(x,n)$  rekursiv funksiya tuzilsin. Bu yerda  $x$  ( $x \neq 0$ ) haqiqiy,  $n$ -butun son.

3. `const int n=40;`  
`float x[n];`  
 Berilgan  $x$  vektorning  $k$  va oxirgi o'rindagi elementlari orasidagi eng kichik elementni topadigan  $\text{min1}(k)$  yordamchi rekursiv funksiya qurilib, uning yordamida  $x$  vektorining eng kichik elementini topadigan  $\text{min}(x)$  funksiyasi tuzilsin.
4. `char satr[100];`  
 Berilgan `satr` satrning  $i$ -elementidan boshlanib,  $j$ -elementida tugaydigan qismini simmetrikligini tekshiradigan rekursiv mantiqiy  $\text{sim}(s,i,j)$  funksiya tuzilsin.
5. `enum Ism=(Sayyora,...,Erkin,null);`  
 Berilgan  $b$  ismli odam  $a$  ismli odamning avlodi (bolasi, nevarasi, chevarasi va hokazo) bo'lishligini tekshiradigan  $\text{avlod}(a,b)$  mantiqiy funksiyasi



- tuzilsin. Masalani yechishda oldindan tuzilgan  $ona(x)$  va  $ota(x)$  funksiyalaridan foydalaning. Bunda  $ona(x)$  va  $ota(x)$  funksiyalari  $x$  ismli odamni mos ravishda onasi va otasining ismini, agar  $x$  ga mos keluvchi ota-onaga to'g'risida ma'lumot bo'lmasa, yo'q (null) qiymatini qabul qiladi.
6. Agarda  $BolalarSoni(x)$  funksiyasi  $x$  ismli odamning farzandlar sonini,  $Bola(x,k)$  funksiyasi esa,  $x$  ismli odamning  $k$ -farzandining ismini bildirsa (bu yerda  $k$  -  $x$  odamning farzandlar sonidan oshmasligi kerak), ushbu funksiyalar yordamida yuqorida keltirilgan masala (4-masala) yechilsin.
  7. Kesmani teng ikkiga bo'lish usuli yordamida  $f(x)=0$  tenglamani  $[a,b]$  oraliqda  $\epsilon$  aniqligidagi ildizini topadigan  $root(f,a,b,\epsilon)$  rekursiv funksiyasi tuzilsin. ( $\epsilon>0, a<b, f(a)*f(b)<0, [a,b]$  oraliqda  $f(x)$  – uzluksiz va monoton funksiya deb hisoblansin.)
  8. O'qish faylida manfiy son bilan tugaydigan, bo'sh bo'lmagan musbat haqiqiy sonlar ketma-ketligi berilgan. Musbat sonlar yig'indisini hisoblaydigan parametrsiz  $sum()$  rekursiv funksiya tuzilsin.
  9. Matndagi (oxiri nuqta bilan tugagan) raqamlar sonini hisoblaydigan parametrsiz  $digits()$  rekursiv funksiya tuzilsin.
  10. O'qish faylida oxiri nuqta bilan tugaydigan matn berilgan. Bu matnni teskari tartibda chop qiladigan  $teskari()$  funksiya tuzilsin.
  11. Nol bilan tugaydigan butun sonlar ketma-ketligi berilgan. Birinchi navbatda ketma-ketlikning barcha manfiy sonlari, so'ngra musbat sonlar chop qilinsin (ixtiyoriy tartibda).
  12. O'qish faylida nuqta bilan tugaydigan matn berilgan. Uning formula ekanligi quyidagi qoidalar asosida tekshirilsin:  
 $\langle formula \rangle ::= \langle raqam \rangle | (\langle formula \rangle \langle belgi \rangle \langle formula \rangle)$   
 $\langle belgi \rangle ::= + | - | *$   
 $\langle raqam \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$  .
  13. O'qish faylida quyidagi ko'rinishda mantiqiy ifoda xatosiz yozilgan.  
 $\langle mantiqiy\ ifoda \rangle ::= true | false | \langle amal \rangle (\langle operandlar \rangle)$   
 $\langle amal \rangle ::= ! | \&\& | ||$   
 $\langle operandlar \rangle ::= \langle operand \rangle | \langle operand \rangle, \langle operandlar \rangle$   
 $\langle operand \rangle ::= \langle mantiqiy\ ifoda \rangle$   
 (&& va || amallarida operandlar soni ixtiyoriy bo'lishi mumkin, ! amalida esa faqat bitta). Bu ifodaning qiymati hisoblansin. (Masalan,  $\&\&(||(false,!(false)), true, !(true)) \rightarrow false$ .)
  14. O'qish faylidan nuqta bilan tugaydigan matn o'qilsin. Uning tuzilishi quyidagi qoidani qanoatlantirishi tekshirilsin.  
 $\langle matn \rangle ::= \langle element \rangle | \langle element \rangle \langle matn \rangle$   
 $\langle element \rangle ::= a | b | (\langle matn \rangle | [\langle matn \rangle] \{ \langle matn \rangle \})$
  15. “Xanoy minorasi” masalasi. Uchta  $A, B, C$  qoziq va  $n$  ta har xil o'lchamli xalqalar mavjud. Xalqalar o'lchamlari o'sish tartibida 1 dan  $n$  gacha tartiblangan. Barcha xalqalar  $A$  qoziqda a) rasmdagidek joylashtirilgan.  $A$  qoziqdagi barcha xalqalarni  $C$  qoziqqa quyidagi qoidalariga amal qilgan holda quyidagi rasmdagidek o'tkazish talab etiladi: xalqalarni bittadan

ko‘chirish kerak va katta o‘lchamli xalqani kichik o‘lchamli xalqa ustiga qo‘ymaslik kerak.



Amallar ketma-ketligini chop etadigan (“*xalqa q dan r ga o‘tkazilsin*” ko‘rinishida, bunda q va r – ‘A’, ‘B’ yoki ‘C’) masalani n ta xalqa uchun yechadigan programma tuzilsin. (ko‘rsatma: xalqalarni A dan C ga to‘g‘ri o‘tkazishda b), d) rasmlardagidek ko‘rinish uchraydi).

16. 5 ta har xil natural son berilgan. Bu sonlarni barcha o‘rin almashish holatlari chop qilinsin.
17. Shaxmat taxtachasida 8 ta farzin shunday joylashtirilgan-ki, ular bir-birini “*urmaydi*”. Barcha shunday holatlar (92 ta) chop etilsin.
18. Berilgan nomanfiy n va m butun sonlar uchun  $A(n,m)$  funksiya qiymati hisoblansin.

$$A(n,m) = \begin{cases} m+1, & n=0; \\ A(n-1,1), & n \neq 0, m=0; \\ A(n-1, A(n,m-1)), & n > 0, m > 0. \end{cases}$$

19. Butun n va n ta har xil elementdan iborat bo‘lgan haqiqiy turdagi vektor berilgan. Quyidagi tez tartiblash usuli yordamida massiv elementlari o‘rish tartibida joylashtirilsin: massivning ixtiyoriy elementini tanlab, (masalan o‘rtadagisini) shu elementning chap tomonida shu elementdan kichik, o‘ng tomonda esa katta elementlari joylashtiriladi. (Bu bilan tanlangan element o‘zining oxirgi joyiga kiradi), keyin shu usul massivning chap va o‘ng qismi uchun rekursiv qo‘llaniladi.
20. 1 dan n gacha nomerlangan n ta aholi punkti mavjud. Ayrim punktlar o‘zaro yo‘llar bilan tutashgan. Bu yo‘llar yordamida 1-punkt dan n-punktga borish mumkin yoki yo‘qligini aniqlansin. Yo‘llar haqidagi ma’lumot i va j ( $i < j$ ) sonlar juftliklari ketma-ketlik ko‘rinishida berilgan. Ketma-ketlik 2 ta nol bilan tugaydi.
21. Ishorasiz butun sonlar massivi a quyidagi usulda tartiblansin. Faraz qilyalik k - sonning kompyuter xotirasidagi ikkilik ko‘rinishidagi razryadlar soni bo‘lsin (masalan, short int turi uchun  $k=16$ ). Massiv elementlari ularning k-razryadi bo‘yicha ikkiga ajratiladi: k-razryadda 0 bo‘lganlari a0 massivga, 1 bo‘lganlari a1 massivga ko‘chiriladi. Xuddi shu usulda a0 va a1 massivlar k-1 razryad bo‘yicha ikkita massivga ajratiladi va hokazo. Jarayon  $k=0$  bo‘lguncha davom etadi va undan keyin jarayon teskari tomonga ketadi va har bir k- qadamdagi a0 va a1 massivlar ulanishi asosida a massiv yangidan shakllanadi va k+1 qadamdan amalga oshirilgan murojaatga natija sifatida qaytariladi.

## 19.Strukturalar

### *Namunaviy masala*

Tekislikda berilgan  $n$  ta  $p_i(x_i, y_i), i = \overline{1, n}$  nuqtalarni o'z ichiga oladigan minimal radiusli aylana aniqlansin.

### *Yechish usuli*

Barcha nuqtalarni o'z ichiga oladigan aylana markazi -  $M(x_m, y_m)$  nuqtalarning geometrik markazi bo'ladi, bu yerda

$$x_m = \frac{\sum_{i=1}^n x_i}{n}, y_m = \frac{\sum_{i=1}^n y_i}{n}.$$

Aylana radiusi sifatida aylana markazidan nuqtalargacha bo'lgan masofalarning eng kattasi olinadi -  $r = \max_{1 \leq i \leq n} \sqrt{(x_m - x_i)^2 + (y_m - y_i)^2}$ .

### *Programma matni*

```
#include <math.h>
#include <iostream.h>
struct Nuqta
{
    float x,y;
};
Nuqta Aylana_Markazi(Nuqta*nuqta,int n)
{
    Nuqta N;
    N.x=0; N.y=0;
    for(int i=0;i<n;i++)
    {
        N.x+=nuqta[i].x; N.y+=nuqta[i].y;
    }
    N.x/=n; N.y/=n;
    return N;
}
float Aylana_radiusi(Nuqta m,Nuqta*nuqta,int n)
{
    float r=0,d;
    for(int i=0;i<n;i++)
    {
        d=sqrt(pow(nuqta[i].x-m.x,2)+pow(nuqta[i].y-m.y,2));
        if(d>r)r=d;
    }
    return d;
}
int main()
```

```

{
int n;
Nuqta a_markaz;
float a_radius;
cout<<"Nuqtalar sonini kiriting: "; cin>>n;
Nuqta *nuqta=new Nuqta[n];
for(int i=0;i<n;i++)
{
cout<<"\nx["<<i+1<<"]="; cin>>nuqta[i].x;
cout<<"y["<<i+1<<"]="; cin>>nuqta[i].y;
}
a_markaz=Aylana_Markazi(nuqta,n);
a_radius=Aylana_radiusi(a_markaz,nuqta,n);
cout<<"Aylana M("<<a_markaz.x<<','<<a_markaz.y<<") ,";
cout<<"R="<<a_radius;
return 0;
}

```

### *Amailiy masalalar*

- enum Rasm={piki,trefi,bubni,cherivi};  
enum Nom={olti,etti,sakkiz,toqqiz,on,valet,dama,qirol,tuz};  
struct Karta{Rasm r;Nom n;};  
Rasm kz;  
Karta k1,k2;  
Zot rasmi kz berilganda k1 karta k2 kartani urishi yoki yo'qligini aniqlovchi mantiqiy Uradi(k1,k2,kz) funksiyasi tuzilsin.
- struct Domino\_tosh{unsigned char chap,ung};  
Domino\_tosh domino[28];  
Berilgan domino qatordagi domino toshlari to'g'ri qo'yilganligini (navbatdagi domino toshining o'ng qismidagi son keyingi toshning chap qismidagi songa tengligini) tekshiradigan Tugri\_qator(domino) mantiqiy funksiya tuzilsin.
- struct Dekard{double x,y};  
struct Qutb{double r,fi}; //  $r \geq 0, -\pi < fi \leq \pi$   
Dekart d;  
Qutb q;  
Berilgan d nuqta koordinatasini dekart (Dekard) koordinatalar sistemasidan qutb (Qutb) koordinatalar sistemasidagi nuqtaga (r) o'tkazadigan DQ(d,r) va teskari almashtirishni bajaruvchi QD(r,d) funksiyalar tuzilsin.
- struct Shaxmat\_maydoni  
{  
char vert; // 'a'..'h'  
unsigned char goriz; // goriz: 1..8  
};  
Shaxmat\_maydoni m1,m2;

- Farzin bir yurishda  $m_1$  maydondan  $m_2$  maydonga o'tishi mumkinligini tekshiradigan `Farzin_yurish(m1,m2)` mantiqiy funksiya tuzilsin.
5. `struct Vaqt{unsigned char soat,min,sek;;  
Vaqt t1,t2;  
Quyidagi masalalar yechilsin:`
- $t_1$  vaqt  $t_2$  vaqtdan oldin kelishini tekshiradigan `oldin(t1,t2)` mantiqiy funksiya tuzilsin (bir sutka ichida);
  - $t$  vaqtdan 1 sekund ortiq vaqtni  $t_1$  parametrga beruvchi `secund(t,t1)` funksiya tuzilsin (sutka almashinuvi hisobga olingan holda);
  - $t_1$  vaqtdan  $t_2$  vaqtgacha qancha vaqt o'tganligini hisoblaydigan `interval(d,t2,t1)` funksiyasi tuzilsin ( $t_2 > t_1$ ).
6. `struct Rasional{int surat, maxraj;;  
Rasional r[20],a,b;  
Quyidagi masalalar yechilsin:`
- $a$  va  $b$  ratsional sonlarning tengligini tekshiruvchi `teng(a,b)` mantiqiy funksiyasi tuzilsin;
  - $a$  va  $b$  ratsional sonlar yig'indisini  $a$  o'zgaruvchisiga qaytaruvchi `summa(a,b)` funksiyasi tuzilsin;
  - $a$  ratsional sonni qisqartirib bo'lmaydigan ko'rinishga keltiradigan `qisqartir(a)` funksiyasi tuzilsin;
  - $r$  ratsional sonlarning eng kattasini  $m$  parametrga beruvchi `max(x,m)` funksiyasi tuzilsin.
7. `struct Kompleks{double re,im;;`  
Kompleks turdagi  $z$  va  $\varepsilon > 0$  haqiqiy sonlari berilgan. Quyidagi funksiyalar  $\varepsilon$  aniqlikda hisoblansin:
- $y = e^z = 1 + \frac{z}{1} + \frac{z^2}{2!} + \dots + \frac{z^n}{n!} + \dots;$
  - $y = \operatorname{sh} z = z + \frac{z^3}{3!} + \frac{z^5}{5!} + \dots + \frac{z^{2n+1}}{(2n+1)!} + \dots;$
  - $y = \cos z = 1 - \frac{z^2}{2!} + \frac{z^4}{4!} - \dots + \frac{(-1)^n z^{2n}}{(2n)!} + \dots;$
  - $y = \ln(1+z) = z - \frac{z^2}{2} + \frac{z^3}{3} - \dots + \frac{(-1)^{n+1} z^n}{n} + \dots, (|z| < 1);$
  - $y = \operatorname{arctg} z = z - \frac{z^3}{3} + \frac{z^5}{5} - \dots + \frac{(-1)^n z^{2n+1}}{(2n+1)} + \dots, (|z| < 1).$
8. `struct Kompleks{double re,im;;  
struct Koeff{Kompleks a,b,c;;// a≠0  
Koeff Kvt;  
Kompleks x1,x2;  
Koeffisiyentlari Kompleks turidagi  $ax^2+bx+c=0$  kvadrat tenglama ildizlari topilsin.`
9. `struct Sana{unsigned char kun,oy;unsigned int yil;;  
enum Hafta_ kuni{dush,sesh,chor,pay,juma,shan,yak};  
Sana d;  
Quyidagi masalalar yechilsin:`
- $d$ -sanaga tegishli bo'lgan oydagi kunlar sonini hisoblovchi `Oy_kunlari(d)` funksiyasi tuzilsin (kabisa bo'lmagan yil uchun);
  - $d$ -sana to'g'riligini (masalan, 31 iyun bo'lmasligi va h.k.) tekshiradigan

mantiqiy Tugri\_sana(d) funksiyasi tuzilsin.

d) 1-yilning 1-yanvaridan to d-sanagacha necha kun o'tganligini hisoblaydigan Kunlarsoni(d) funksiyasi tuzilsin.

e) d - sana haftani qaysi kunini aniqlaydigan Hafta\_kuni(d) funksiyasi tuzilsin (eramizning 1-yilining 1-yanvari dushanba (dush) kuni bo'lganligini hisobga olgan holda).

10. struct Manzil{char Kucha[20]; int uy,xonadon;};  
struct Yashovchi{char familiya[20], shahar[20]; Manzil manzil;};  
Yashovchi \*shaxs;  
Berilgan shaxs ro'yxatidagi turli shaharda, lekin bir xil manzilda (Manzil) yashovchi ixtiyoriy ikkita shahar yashovchisining familiyasini chop qiladigan Taqdir\_xazili() funksiyasi tuzilsin.
11. O'qish faylida talabalarning sessiya natijalari haqidagi ma'lumot quyidagi ko'rinishga ega:  
<familiya>,<guruh nomeri>,<reyting1>,<reyting2>,<reyting3>.  
Bu yerda familiya 15 harfgacha bo'lgan satr, guruh nomeri - butun son, har bir reyting 0 va 100 oraligidagi haqiqiy son: reyting1- matematik analizdan, reyting2 - algebradan, reyting3- programmalash predmetlaridan. Quyidagilar aniqlansin:  
a) kamida bitta fandan qarzidor bo'lgan talabaning familiyasi;  
b) barcha imtihonlarni 4 va 5 bahoga topshirgan talabalar necha foizni tashkil qilishi;  
d) talabalar qaysi fandan imtihonlarni eng yaxshi natija bilan topshirganligi;  
e) talabalar o'zlashtirishining o'rtacha qiymati bo'yicha o'smaydigan tartibdagi guruhlarining tartib nomerlari.
12. O'qish faylida oliy o'quv yurti talabalari to'g'risidagi quyidagi ma'lumot yozilgan:  
<familiya>,<ismi>,<otasining ismi>,<jinsi>,<yoshi>,<kursi>.  
Bu yerda familiya, ismi va otasining ismi – har biri uzunligi 12 harfdan ko'p bo'lmagan satrlar, jinsi - 'e' yoki 'a' harflar bilan (erkak, ayol) ko'rsatilgan, yoshi - 16 dan 35 gacha bo'lgan butun son, kursi - 1 dan 4 gacha bo'lgan butun son. Quyidagi masalalar yechilsin:  
a) erkaklar soni eng ko'p bo'lgan kurs nomeri;  
b) eng ko'p tarqalgan erkak va ayollar ismlari;  
d) yoshi va ismi bir xil bo'lgan talaba qizlar familiyalarining alfavit tartibidagi ro'yxati.
13. AEROFLOT strukturasi <reysning manzil punkti>,<reys nomeri> va <samolet turi> maydonlarini o'z ichiga oladi. Berilgan n o'lchamidagi AEROFLOT turidagi massiv yaratilib, ma'lumotlar kiritilsin va quyidagi amallar bajarilsin:  
a) massiv, reys nomerining o'sichi bo'yicha tartiblansin;  
b) klaviaturadan kiritilgan manzilga uchadigan reys nomeri va samolet turi chop etilsin;  
d) klaviaturadan kiritilgan samolet turi uchadigan reys nomerlari va manzillari

chop etilsin.

14. **ISHCHI** strukturasi <familiya va initsiallari>,<lavozimi> va <ishga kirgan yili> maydonlarini o'z ichiga oladi. Berilgan n o'lchamidagi ISHCHI turidagi massiv yaratilsin va quyidagi amallar bajarilsin:
  - a) massiv ishchi familiyasining alfavit bo'yicha joylashuviga mos tartiblansin;
  - b) klaviaturadan kiritilgan yildan keyin ishga kirgan ishchilar familiyasi chop etilsin;
  - d) klaviaturadan kiritilgan lavozimdagi ishchilar familiyasi va ishga kirgan yili chop etilsin.
15. **POEZD** strukturasi <poezd nomeri>,<boradigan manzil nomi> va <jo'nash vaqti> maydonlarini o'z ichiga oladi. Berilgan n o'lchamidagi POEZD turidagi massiv yaratilsin va quyidagi amallar bajarilsin:
  - a) massiv poyezd nomerlarining o'sishi bo'yicha tartiblansin;
  - b) klaviaturadan kiritilgan nomeriga mos poezd nomeri haqidagi ma'lumot chop etilsin;
  - d) klaviaturadan kiritilgan manzilga va ko'rsatilgan vaqtdan keyin jo'naydigan poezdlar haqidagi ma'lumot chop etilsin.
16. **MARSHRUT** strukturasi <marshrut boshi>,<marshrut oxiri> va <marshrut nomeri> maydonlarini o'z ichiga oladi. Berilgan n o'lchamidagi MARSHRUT turidagi massiv yaratilsin va quyidagi amallar bajarilsin:
  - a) massiv marshrut nomerlarining o'sishi bo'yicha tartiblansin;
  - b) klaviaturadan nomeri kiritilgan marshrut haqidagi ma'lumot chop etilsin;
  - d) klaviaturadan kiritilgan nom bo'yicha, boshlanashi yoki oxiri shu nomdagi punkt bo'lgan marshrutlar haqidagi ma'lumot chop etilsin.
17. **BLOKNOT** strukturasi shaxs haqidagi - <familiya va ismi>,<telefon nomeri> va <tug'ilgan sanasi> maydonlarini o'z ichiga oladi. Berilgan n o'lchamidagi BLOKNOT turidagi massiv yaratilsin va quyidagi amallar bajarilsin:
  - a) massiv odamning tug'ilgan sanasining o'sishi bo'yicha tartiblansin;
  - b) klaviaturadan telefon nomeri kiritilgan odam haqidagi ma'lumot chop etilsin;
  - d) tug'ilgan oyi klaviaturadan kiritilgan songa mos keluvchi odamlar haqidagi ma'lumot chop etilsin.
18. **NARX** strukturasi mahsulot haqidagi <mahsulot nomi>,<mahsulot sotiladigan magazin nomi> va <mahsulotning so'mdagi narxi> maydonlarini o'z ichiga oladi. Berilgan n o'lchamidagi NARX turidagi massiv yaratilsin va quyidagi amallar bajarilsin:
  - a) massiv mahsulot nomini alfavit bo'yicha joylashuviga mos tartiblansin;
  - b) nomi klaviaturadan kiritilgan mahsulot haqidagi ma'lumot chop etilsin;
  - d) klaviaturadan kiritilgan nomdagi magazinda sotiladigan mahsulotlar ro'xati va ularning jami narxi chop etilsin.
19. **ORDER** strukturasi bank mijozi haqidagi <to'lovchining hisob raqami>,<oluvchining hisob raqami> va <o'tkaziladigan pul miqdori> maydonlarini o'z ichiga oladi. Berilgan n o'lchamidagi ORDER turidagi

massiv yaratilsin va quyidagi amallar bajarilsin:

- a) massiv to'lovchining hisob raqami bo'yicha tartiblanisin;
- b) hisob raqami klaviaturadan kiritilgan to'lovchining hisob raqamidan qancha pul olinganligi haqidagi ma'lumotlar chop etilsin;
- d) hisob raqami klaviaturadan kiritilgan oluvchining hisob raqamiga qaysi hisobdan qancha pul kelib tushganligi haqidagi ma'lumot chop etilsin.

## 20. Matn fayli

### *Namunaviy masala*

$f_1$  va  $f_2$  matn fayllarida butun sonlar kamaymaydigan tartibda kiritilgan. Ushbu fayllar elementlaridan, sonlarning tartiblanishi saqlangan holda,  $f_3$  matn fayli hosil qilinsin. Fayllarning diskdagi nomlari buyruq satri orqali kiritilsin.

### *Yechish usuli*

Fayllar bilan ishlash uchun <iostream.h> kutubxonasida aniqlangan funksiyalardan foydalanamiz:

**fopen()** – fayl o'zgaruvchisini diskdagi fizik fayl bilan bog'lash uchun xizmat qiladi. Agar funksiya **NULL(0)** qiymatini qaytarsa, bog'lanish muvaffaqiyatsiz bo'lganligini bildiradi;

**fscanf()** funksiyasidan fayldagi sonni formatli o'qish va fayl tugaganligini aniqlash uchun foydalaniladi. Funksiyaning **-1(EOF)** qiymatni qaytarishi, fayl tugaganligini anglatadi.

Masalani yechish algoritmi quyidagicha:

1.  $f_1$  va  $f_2$  fayldan mos ravishda **a, b** sonlari o'qiladi;
2. Toki  $a \leq b$  va  $f_1$  tugamaganlik sharti bajarilar ekan **a** qiymati  $f_3$  fayliga yozilsin va  $f_1$  fayldan qiymat **a** o'zgaruvchiga o'qilsin;
3. Toki  $b > a$  va  $f_2$  tugamaganlik sharti bajarilar ekan **b** qiymati  $f_3$  fayliga yozilsin va  $f_2$  fayldan qiymat **b** o'zgaruvchiga o'qilsin;
4. O'qilayotgan fayllardan kamida bittasi tugamagan bo'lsa 2-qadamga o'tilsin.
5. Tamom.

Fayllar nomlari buyruq satrida programma nomidan ("*exe*" fayldan) keyin probel ajratuvchisi bilan yoziladi va ular **main()** funksiyasining **argv** – satrlar massiviga joylashadi, **argc** - buyruq satrdagi satrostilari soni. Masalan,

"d:\f1\_f2\_f3.exe file1.txt file2.txt file3.txt"  
buyruq satrida 4 ta satrostilari mavjud.

C++Builder muhitidagi **Run→Parameters** Parametrs maydoniga "file1.txt file2.txt file3.txt" satrini yozish orqali buyruq parametrlarini berish va programmani ishga tushirish mumkin.

### *Programma manti*

```
const int n=80;
```



```

typedef FILE * Fmatn;
typedef char Satr[n];
int Ulash(Satr f1_nomi,Satr f2_nomi,Satr f3_nomi)
{
    int a,b;
    bool tamom_f1=false,tamom_f2=false;
    Fmatn f1,f2,f3;
    if((f1=fopen(f1_nomi,"rt"))==NULL)
    {
        cout<<" "<<f1_nomi<<" fayli ochilmadi!";
        return 1;
    }
    if((f2=fopen(f2_nomi,"rt"))==NULL)
    {
        cout<<" "<<f2_nomi<<" fayli ochilmadi!";
        return 1;
    }
    if((f3=fopen(f3_nomi,"wt"))==NULL)
    {
        cout<<" "<<f3_nomi<<" fayli hosil qilinmadi!";
        return 1;
    }
    if(fscanf(f1,"%i",&a)==EOF) tamom_f1=true;
    if(fscanf(f2,"%i",&b)==EOF) tamom_f2=true;
    do
    {
        while(!tamom_f1&&(tamom_f2||a<=b))
        {
            fprintf(f3,"%i ",a);
            if(fscanf(f1,"%i",&a)==EOF) tamom_f1=true;
        }
        while(!tamom_f2&&(tamom_f1||b<a))
        {
            fprintf(f3,"%i ",b);
            if(fscanf(f2,"%i",&b)==EOF) tamom_f2=true;
        }
    }
    while (!tamom_f1||!tamom_f2);
    fclose(f1);
    fclose(f2);
    fclose(f3);
    return 0;
}
int main(int argc,char* argv[])

```

```

{
  Satr f1_nomi,f2_nomi,f3_nomi;
  if(argc!=4)
  {
    cout<<"Buyruq satri noto'g'ri kiritilgan!";
    return 0;
  }
  strcpy(f1_nomi,argv[1]);
  strcpy(f2_nomi,argv[2]);
  strcpy(f3_nomi,argv[3]);
  if(!Ulash(f1_nomi,f2_nomi,f3_nomi))
    cout<<"\nFayl hosil qilindi: f1+f2=>f3";
  else
    cout<<"\nMasalani echish imkoniyati yo'q.";
  return 0;
}

```

### ***Amamliy topshiriqlar***

1. Berilgan **s** matn faylning manfiy elementlari yig'indisini hisoblaydigan funksiya tuzilsin.
2. Bo'sh bo'lmagan, butun sonlardan tashkil topgan **r** matn fayldagi sonlar o'sish yoki kamayish tartibida joylashganligini tekshiradigan mantiqiy funksiya tuzilsin.
3. Berilgan **t<sub>1</sub>** va **t<sub>2</sub>** matn fayllar biri ikkinchisining nusxasi yoki yo'qligini tekshiradigan mantiqiy funksiya tuzilsin.
4. Kamida ikkita satrdan iborat bo'lgan **f** matn faylining oxiridan bitta oldingi satrini natija sifatida qaytaradigan funksiya tuzilsin.
5. Berilgan **s** satrda uchragan raqamlarni **t** matnga o'tkazadigan funksiya tuzilsin.
6. Berilgan musbat butun **n** sonidan oshmaydigan Fibonachchi sonlarini **f** matn fayliga yozadigan funksiya tuzilsin.
7. Butun sonlar yozilgan, bo'sh bo'lmagan **f** matn fayl berilgan. Fayldagi elementlarning o'rta arifmetigidan kichik bo'lgan fayl elementlari miqdorini aniqlaydigan funksiya tuzilsin.
8. Berilgan **f** matn fayli bo'sh bo'lmagan **n** satrdan iborat. Satrdagi so'zlar bir-biridan vergul bilan ajratilgan, oxirgi so'z nuqta bilan tugagan. Har bir satrning oxirgi so'zidan farq qiluvchi so'zlardan tashkil topgan satr **g** matn faylga yozilsin.
9. Berilgan **t** matn fayl uchun quyidagi funksiyalar tuzilsin:
  - a) **add1(t,s)**, **t**-matn boshiga **s** satrni qo'shadigan;
  - b) **addlast(t,s)**, **t**-matn oxiriga **s** satrini qo'shadigan;
  - d) **double(t)**, **t**-matndagi har bir raqamni ikkilantiradigan;
  - e) **replace(t,s)** - bo'sh bo'lmagan **t** matnning oxirgi satrini **s** satri bilan

almashtiradigan;

f) **next(t)** - **t** matnda uchragan har bir raqamni, shu raqamdan keyin keluvchi raqam bilan almashtiradigan ('9' raqami '0' bilan almashtiriladi);

j) **del(t)**- **t** matndagi oxiridan bitta oldingi satrini (agar u mavjud bo'lsa) o'chiradigan;

i) **first(t)** - **t** matnda har bir satrning faqat birinchi uchraganini qoldiradigan.

10. Haqiqiy sonlar yozilgan **f** matn faylidagi eng uzun o'suvchi ketma-ketlik elementlari miqdorini aniqlovchi funksiya tuzilsin.

11. Berilgan **f** va **g** matn fayllarida sonlar kamaymaydigan ko'rinishda tartiblangan bo'lsin. Bu fayllarni yagona kamaymaydigan ko'rinishda **h** faylga birlashtirish talab qilinadi.

12. Mantiqiy **relation(f)** funksiyasi tuzilsin. Bu funksiya **f** fayldagi berilganlar "*munosabat*" sintaksisidagi to'g'ri yozuv ekanligi aniqlasin (qoidalar quyida keltirilgan).

**<munosabat>::=<son><munosabat belgisi><son>**

**<munosabat belgisi>::=<|=|>|<=|<>|>=**

**<son>::=<raqam>|<raqamlar>**

**<raqamlar>::=<nol emas><raqam>|<raqamlar><raqam>**

**<raqam>::=0|<nol emas>**

**<nol emas>::=1|2|3|4|5|6|7|8|9**

13. To'qqizta satrlardan iborat **t** matn faylini hosil qiluvchi funksiya tuzilsin. Bunda 1-satrdagi bitta '1' belgisi, 2-satrdagi ikkita '2' belgisi va shu tariqa 9-satrdagi to'qqizta '9' belgisi bo'lsin.

14. Klaviaturadan belgilarni bittalab, birinchi nuqtagacha o'qiydigan va ularni **t** faylga 40 ta belgidan iborat satr ko'rinishida yozadigan funksiya tuzilsin (satrga nuqta kirmaydi va oxirgi satrdagi belgilar 40 dan kam bo'lishi mumkin).

15. Quyidagilarni amalga oshiradigan funksiya tuzilsin:

a) **t** matn fayldagi bo'sh satrlar sonini hisoblaydigan;

b) **t** matn faylidagi satrlar ichida eng uzun satr uzunligini hisoblaydigan.

16. Berilgan **t** matn fayli bo'sh bo'lmagan satrlardan iborat bo'lsin. Quyidagi shartlarni bajaruvchi satrlar sonini hisoblovchi funksiya tuzilsin:

a) 'd' harfidan boshlanadigan;

b) 'z' harfi bilan tugaydigan;

d) bir xil belgi bilan boshlanadigan va tugaydigan;

e) bir xil belgilardan tashkil topgan.

17. Berilgan **t<sub>1</sub>** matn faylidagi satrlarga bo'linishini saqlagan holda **t<sub>2</sub>** faylga o'tkazadigan funksiya tuzilsin.

18. Berilgan **t** matn fayli har birining uzunligi 80 belgidan oshmaydigan satrlarga bo'lingan deb hisoblab funksiya tuzilsin. Funksiya **t** fayldagi har bir satrni 80 belgigacha o'ng tomondan probel ('□') bilan to'ldirib, barcha satrlari 80 belgidan iborat **t<sub>2</sub>** faylga o'tkazadi.

19. **char suz[ ][20];**

Belgilar soni 20 tagacha bo'lgan sozlar ro'yxati berilgan (**suz**). Har bir so'zni **t** matn fayliga alohida satr ko'rinishida o'tkazilsin.

20. Matn  $t$  faylida bo'sh bo'lmagan va probel bilan ajratilgan haqiqiy sonlar ketma-ketligi yozilgan. Bu sonlar ichida eng kattasini topadigan funksiya tuzilsin.
21. Berilgan  $t_1$  matn faylida bo'sh bo'lmagan va probel bilan ajratilgan butun sonlar ketma-ketligi yozilgan. Bu sonlar ichida musbatlarini  $t_2$  faylga o'tkazadigan funksiya tuzilsin.
22. Bo'sh bo'lmagan  $t$  matn faylidagi har satrni, uning boshiga satrning tartib nomerini qo'yib chop qiladigan funksiya tuzilsin. Satr tartib nomeri 4 o'rin egallaydi va undan keyin probel qo'yiladi.
23. Berilgan Kitob matn faylidagi satrlarning boshlang'ich bo'linishlarini inkor etgan holda shunday satrlarga bo'linsinki, natijada satr nuqta bilan tugasin yoki 60 belgidan iborat bo'lsin, agar bu belgilar orasida nuqta uchramasa.
24. Berilgan matn faylidagi eng qisqa satrlarning birinchisi chop qilinsin.
25. Matn fayli va  $s$  satr berilgan.  $s$  satr fayl boshiga (oxiriga) qo'shilsin.
26. Berilgan matn fayldan birinchi (oxirgi) satr o'chirilsin.
27. Name1 va Name2 nomli matn fayllari berilgan. Ulardagi satrlarni ketma-ket birlashuvidan yangi Name3 fayli hosil qilinsin. Fayllarni birlashish tartibi foydalanuvchi tomonidan kiritiladi.
28. Matn fayli va  $k$  butun soni berilgan. Matn fayldagi  $k$ -satr o'chirilsin. Agar faylda bu satr bo'lmasa, u o'zgarmasdan qoldirilsin.
29. Matn fayli va  $k$  butun soni berilgan. Matn fayldagi  $k$ -satr oldiga (keyinga) bo'sh satr qo'yilsin. Agar faylda bu satr bo'lmasa u o'zgarmasdan qoldirilsin.
30. Matn fayli va  $s$  satri berilgan. Fayldagi barcha bo'sh satrlar  $s$  satr bilan almashtirilsin.
31. Matn fayli berilgan. Undagi ketma-ket keluvchi probellar bitta probel bilan almashtirilsin.
32. Ichidagi satrlari 60 belgidan oshmagan va chap tomonga tekislangan matn fayli berilgan. Har bir bo'sh bo'lmagan satrlar oldiga etarli sondagi probelni qo'yish orqali satrlar o'ng tomonga (markazga) tekislansin.
33. Berilgan matn faylidagi abzats boshlanishini 5 ta probeldan boshlanuvchi satr aniqlaydi. Matn fayli abzats oldidagi probellar olib tashlash va satr oldiga bo'sh satrni qo'yish orqali yangi ko'rinishga keltirilsin.
34. Berilgan matn faylidagi satrlarni teskari tartibda joylashtirish orqali yangi matn fayli hosil qilinsin.
35. Matn fayli va butun  $k$  soni berilgan. Matn faylidagi  $k$ - abzats o'chirilsin. Abzats boshida 5 ta probel bilan boshlanadigan satr bilan aniqlanadi. Agar bu tartib nomerli abzats bo'lmasa fayl o'zgarishsiz qoldirilsin.
36. Har biri chap va o'ng tomondan probellar bilan to'ldirilgan haqiqiy sonni ifodalovchi satrlardan tashkil topgan matn fayli berilgan. Bu sonlarning yig'indisi va miqdori aniqlansin. Masalani yechishda faqat fayldan belgili o'qish amalga oshirilsin.
37. Name1 va Name2 nomli matn fayllari berilgan. Ulardagi mos tartib nomerdagi satrlarni ketma-ket yozishdan yangi Name3 fayli hosil qilinsin (Name1 faylining 1-satri, Name2 faylining 1-satri, Name1 faylining 2-satri,

- Name2 faylining 2-satri va hokazo). Agar birorta fayl ikkinchisidan qisqa bo'lsa, u tugashi bilan ikkinchi faylning qolgan satrlari Name3 fayliga ko'chiriladi.
38. Ikkita matn fayli berilgan. Ularning faqat bittasiga kiruvchi satrlar chop etilsin.
  39. Probel bilan ajratilgan, uchtagacha son yozilgan satrlardan iborat Name fayli berilgan. Yangi Name1, Name2 va Name3 matn fayllari hosil qilinsinki, ular mos ravishda har bir satrdagi uchta sonlarning yozuvlaridan (satrostilaridan) tashkil topsin. Agar Name fayli satrida mos o'rinda son bo'lmasa (\* yozilgan), ular o'rniga faylga "NULL" satri yozilsin.
  40. Name1 nomli bo'sh bo'lmagan matn fayli va k natural soni berilgan. Ikkita matn fayllari hosil qilinsin: Name2 faylga Name1 har bir satrning birinchi (oxirgi) k ta belgisini o'zida saqlovchi, (agarda satr uzunligi k kichik bo'lsa satrning o'zini saqlasin), Name3 fayli esa har bir satrning k - belgisidan tashkil topsin (agarda satr uzunligi k dan kichik bo'lsa Name3 fayliga probel yozilsin).
  41. Ikkita  $f_1$  va  $f_2$  matn fayllarini satrma-satr solishtiradigan funksiya tuzilsin. Funksiya  $f_2$  fayldagi  $f_1$  faylga nisbatan o'chirilgan yoki qo'shilgan satrlarni chop qilsin.
  42. C++ tilidagi programmadagi har bir identifikator va u uchragan satrga ko'rsatgich eslab qolish orqali identifikatorlar va kalit so'zlari lug'ati yaratilsin.
  43. Tekislikda to'g'ri chiziq  $ax+by+c=0$  tenglama bilan beriladi. Bunda  $a, b$  koeffisientlari bir vaqtda 0 teng emas va  $a, b, c$  - butun sonlar. Agar f faylda to'g'ri chiziqlar koeffisientlari saqlangan bo'lsa (uchdan kam bo'lmagan), f fayldan g faylga quidagi shartlarni bajaruvchi to'g'ri chiziqlar koiffisientlarini yozing:
    - a) birinchi to'g'ri chiziqqa parallel;
    - b) birinchi to'g'ri chiziqqa parallel va bir-biridan farqli chiziqlar;
    - d) berilgan to'g'ri chiziq bilan kesishadigan chiziqlar.
  44. Butun sonlardan iborat mant fayl berilgan, shu fayl elementlarining teskari tartibda joylashuvidan hosil bo'lgan yangi fayl hosil qilinsin.
  45. Elementlari haqiqiy sonlardan iborat kvadrat matritsani satrlar bo'yicha o'zida saqlovchi matn fayl berilgan. Matritsaning i-satr va j-ustundagi elementlari chop qilinsin. Agar matritsaning i-satri yoki j-ustuni bo'lmasa, bu holda xabar berilsin.
  46. Lotin harflaridan tashkil topgan satr berilgan. Undagi harflarning necha marta uchraganligini ifodalovchi gistogramma matn faylda (son orqali) hosil qilinsin.

## 21. Binar fayllar

### *Namunaviy masala*

Talaba haqidagi ma'lumot uning familiyasi va initsiallari, o'quv bosqichi, fandan boyicha reyting bali bilan berilgan. Talabalar haqidagi ma'lumot binar

fayliga saqlansin. Fayldan ko'rsatilgan baho olgan talabalar ro'yxati chop etilsin.

### ***Yechish usuli***

Har bir talaba haqidagi ma'lumot matn faylidan o'qilib <familiyasi va initsiallari>, <o'quv bosqichi > va <fandan boyicha reyting bali> maydonlaridan tashkil topgan struktura turidagi o'zgaruvchiga o'zlashtiriladi va binar faylga yoziladi. Berilganlar tugagandan keyin binar fayl yopiladi. Klaviaturadan 2 va 5 oralig'dagi baho soni kiritiladi va binar fayl o'qish regimida qayta ochiladi va talaba haqidagi berilganlar strukturaga o'qiladi. Agar strukturaning <fandan boyicha reyting bali> maydoni qiymati baho qiymatiga mos oraliqqa tegishli bo'lsa, struktura maydonlarining qiymatlari ekranga chop qilinadi. Ushbu jarayon binar fayl tugaguncha davom etadi.

O'qishni yaxlit satr sifatida qabul qilish uchun talaba familiya va initsiallari '.' belgisi bilan ajratilib kiritiladi.

### ***Programma matni***

```
const int N=80;
typedef char Fayl_nomi[N];
struct Talaba
{
    char FISh[30];
    unsigned char kurs;
    float ball;
};
int Talaba_Matn_Binar(FILE *t, FILE *b)
{
    int n=0;
    Talaba talaba;
    do
    {
        if(fscanf(t,"%s %i %f", &talaba.FISh, &talaba.kurs, &talaba.ball)==EOF)
            break;
        else
        {
            n++;
            if(fwrite(&talaba,sizeof(Talaba),1,b)!=1)
            {
                printf("Binar faylga yozishda xatolik bo'ldi!");
                return -1;
            }
        }
    }
    return n;
}
```

```

int Bahoga_mos_Talabalar(FILE * bfile, int baho)
{
    int n=0;
    Talaba talaba;
    bool talaba_bor, sarlavha=true;
    while(1)
    {
        if(!fread(&talaba,sizeof(Talaba),1,bfile)) break;
        talaba_bor=false;
        switch(baho)
        {
            case 5: talaba_bor=talaba.ball>85; break;
            case 4: talaba_bor=(talaba.ball>71 && talaba.ball<86); break;
            case 3: talaba_bor=(talaba.ball>55 && talaba.ball<=71); break;
            default: talaba_bor=(talaba.ball<56);
        }
        if(talaba_bor)
        {
            n++;
            if(sarlavha)
            {
                printf(" Fandan bahosi %i bo'lgan talabalar ro'yxati:\n",baho);
                sarlavha=false;
            }
            printf("FISH:%s          Bosqich:%i          Reyting
Ball:%6.2f\n",talaba.FISh,talaba.kurs, talaba.ball);
        }
    }
    return n;
}

int main(int argc, char* argv[])
{
    int n;
    unsigned char baho;
    FILE * f_matn,*f_binar;
    Fayl_nomi mf_nomi,bf_nomi;
    printf("Matn fayl nomini kiriting:");
    scanf("%s",&mf_nomi);
    printf("Binar fayl nomini kiriting:");
    scanf("%s",&bf_nomi);
    if((f_matn=fopen(mf_nomi,"rt"))==NULL)
    {
        printf("%s fayli ochilmadi!",mf_nomi);
        return -1;
    }
}

```

```

}
if((f_binar=fopen(bf_nomi,"wb"))==NULL)
{
printf("%s faylni yaratishda xatolik ro'y berdi!",bf_nomi);
return -1;
}
do
{
printf("Bahoni kiritng(2..5):");
scanf("%i",&baho);
}
while(baho<2 || baho>5);
n=Talaba_Matn_Binar(f_matn, f_binar);
if(n)
{
fclose(f_binar);
f_binar=fopen(bf_nomi,"rb");
n=Bahoga_mos_Talabalar(f_binar,baho);
if(n)printf("Jami talabalar soni:%i",n);
else printf("%i baho olgan talabalar yo'q",baho);
}
fclose(f_matn);
fclose(f_binar);
return 0;
}

```

### ***Amaliy topshiriqlar***

1. SEMESTR faylida 1-bosqich o'quv semestrining natijalari haqida ma'lumot mavjud. Har bir talaba to'g'risidagi ma'lumot quyidagi ko'rinishda berilgan:  
<familiya>,<guruh nomeri>,<baho1>,<baho2>,<baho3>.  
Bu yerda familiya - 12 harfgacha, guruh nomeri - 101 dan 116 gacha butun son, har bir baho- 2,3,4 yoki 5, baho1- matematik analizdan, baho2- algebradan, baho3- programmalashdan. Quyidagi natijalarni chop qiluvchi funksiya tuzilsin:
  - a) kamida bitta fandan qarzidor bo'lgan talabaning familiyasi;
  - b) barcha imtihonlarni 4 va 5 ga topshirgan talabalar necha foizni tashkil qilishini;
  - d) talabalar qaysi fandan imtihonlarni eng yaxshi natija bilan topshirganlar;
  - e) talabalar o'zlashtirishining o'rtacha qiymati bo'yicha o'smaydigan tartibdagi guruhlarning tartib nomerlari.
2. ANKETA faylida oliy o'quv yurti talabalarining har biri to'g'risida quyidagi ma'lumot yozilgan:  
<familiya>,<ismi>,<sharifi>,<jinsi>,<yoshi>,<bosqich>,  
bu yerda talabaning familiyasi, ismi va otasining ismi -satrlar, jinsi 'E' va 'A'



- harflar bilan ko'rsatilgan, yoshi - 16 dan 35 gacha bo'lgan butun son, o'quv bosqich - 1 dan 4 gacha bo'lgan butun son. Quyidagi shartlar uchun natija beruvchi funksiya tuzilsin:
- a) erkaklar soni eng ko'p bo'lgan kurs nomeri;
  - b) eng ko'p tarqalgan erkak va ayollar ismlari;
  - d) yoshi va shariflari bir vaqtda eng ko'p tarqalgan talaba qizlar familiyalarining alfavit tartibidagi ro'yxati.
3. **AEROFLOT** strukturasi, <reysning manzil punkti>, <reys nomeri>, <samolet turi> va <bo'sh o'rinlar soni> maydonlarini o'z ichiga oladi. Ushbu strukturadagi berilganlar f binar faylga yozilgan. f binar fayl ustida quyidagi amallar bajarilsin:
- a) klaviaturadan kiritilgan manzilga uchadigan reys nomeri va samolet turi chop etilsin;
  - b) klaviaturadan kiritilgan samolet turi uchadigan reys nomerlari va manzillar chop etilsin.
  - d) ko'rsatilgan manzil uchun bo'sh o'rinlar sonlari ko'rsatilsin.
4. **POEZD** strukturasi, <poezd nomeri>, <manzil nomi>, <jo'nash vaqti> va <bo'sh o'rinlar> maydonlarini o'z ichiga oladi. Ushbu strukturadagi berilganlar f binar faylga yozilgan. f binar fayl ustida quyidagi amallar bajarilsin:
- a) klaviaturadan kiritilgan poezd nomeriga mos poezd haqidagi ma'lumot chop etilsin;
  - b) klaviaturadan kiritilgan manzilga va ko'rsatilgan vaqtdan keyin jo'naydigan poezd haqidagi ma'lumot chop etilsin.
  - d) ko'rsatilgan poezd nomerida bo'sh o'rinlar bo'lsa uning qiymati berilgan k soniga kamaytirilsin.
5. **BLOKNOT** strukturasi tanish odamning <familiya va ismi>, <telefon nomeri> va <tug'ilgan sanasi> maydonlarini o'z ichiga oladi. Ushbu strukturadagi berilganlar f binar faylga yozilgan. f binar fayl ustida quyidagi amallar bajarilsin:
- a) klaviaturadan telefon nomeri kiritilgan odam haqidagi ma'lumot chop etilsin;
  - b) tug'ilgan oyi klaviaturadan kiritilgan songa mos keluvchi odamlar haqidagi ma'lumot chop etilsin;
  - d) klaviaturadan kiritilgan familiya va ism bo'yicha f fayldan yozuv o'chirilsin.
6. **MUCHAL** strukturasi shaxsning <familiyasi va ismi>, <muchal nomi> va <tug'ilgan sanasi> maydonlarini o'z ichiga oladi. Ushbu strukturadagi berilganlar f binar faylga yozilgan. f binar fayl ustida quyidagi amallar bajarilsin:
- a) massiv shaxsning tug'ilgan sanasini o'sishi bo'yicha zanjir ko'rinishidagi dinamik struktura yordamida tartiblansin va g binar faylga yozilsin;
  - b) familiyasi klaviaturadan kiritilgan familiya bilan ustma-ust tushadigan shaxslar haqidagi ma'lumot chop etilsin;

- d) klaviaturadan kiritilgan muchal yili tug‘lgan shaxslar haqidagi ma’lumot chop etilsin.
7. **BANK** strukturasi bank mijozlari haqidagi <mijoz familiya,ismi va sharifi>,<hisob raqami>,<mablag‘ miqdori>,<foyiz>,<mablag qo‘yilgan yil>,<mijoz manzili> maydonlarni o‘z ichiga oladi. Ushbu strukturadagi berilganlar f binar faylga yozilgan. f binar fayl ustida quyidagi amallar bajarilsin:
    - a) dinamik zanjir strukturasiidan foydalangan holda mijoz familiyasining alfavit bo‘yicha tartiblangan ro‘yxati chop etilsin;
    - b) klaviaturadan hisob raqami ko‘rsatilgan mijoz hisobiga, uning foyiziga mos ravishda qo‘shiladigan pul miqdori hisoblansin;
    - d) klaviaturadan kiritilgan yilda bankka mablag‘ qo‘ygan mig‘oz haqidagi ma’lumot chop etilsin.
  8. **KIYIM** fayli <mahsulot shifri>,<o‘lcham>,<bo‘yi>,<rega bo‘yicha mahsulot soni>,<amalda tayyor mahsulotlar soni> maydonlaridan iborat. **MAHSULOT** fayli ushbu mahsulotlar haqidagi <mahsulot shifri>,<mahsulot nomi> maydonlaridan iborat strukturadagi berilganlardan tashkil topgan. Ekranda <mahsulot shifri>,<mahsulot nomi>,<o‘lcham>,<bo‘yi>,<rega bo‘yicha mahsulot soni>,<amalda tayyor mahsulotlar soni> ustunlaridan iborat jadval chop etilsin.
  9. **TALABA** faylida <fakultet shifri>,<o‘quv kursi>,<guruh nomeri>,<talaba familiyasi va sharifi>,<predmet shifri>,<predmet bo‘yicha baho> maydonlaridan iborat. **FAKULTET** fayli fakultetlar haqidagi <fakultet shifri>,<fakultet nomi> maydonlardan tashkil topgan. **PREDMET** faylida esa <fakultet shifri>,<o‘quv kursi>,<predmet shifri>,<predmet nomi> berilganlari joylashgan. Ekranga har bir o‘quv predmeti bo‘yicha o‘zlashtirish jadvali chop etilsin. Unda <fakultet nomi>,<o‘quv kursi>,<guruh nomeri>,<talaba familiyasi va sharifi>,<predmet bo‘yicha baho> ustunlari bo‘lsin.
  10. **SHAXS** faylida shaxs haqidagi <familiya va sharifi>,<tugilgan yili>,<ma’lumoti>,<lavozimi> maydonlaridan, **ISH\_HAQI** fayli ish haqi bo‘yicha <lavozim>,<ish haqi> maydonlaridan iborat. Ekranga korxona xodimlari haqidagi ma’lumot beruvchi va <familiya va sharifi>,<tugilgan yili>,<ma’lumoti>,<lavozimi> va <ish haqi> ustunlaridan tashkil topgan jadval chop etilsin.
  11. **DARSXONA** fayli <dars turi>,<auditoriya nomeri>,<auditoriya sig‘imi> maydonlaridan tashkil topgan. **DARS** fayli esa <predmet>,<dars turi>,<talabalar soni> maydonlardan iborat. **DARS** faylidagi har bir predmet uchun unga mos keluvchi auditoriya nomerlari chop etilsin.
  12. **KUTUBXONA** fayli <kitob registratsiya nomeri>,<muallif>,<kitob nomi>,<nashr yili> va <nashriyot> maydonlaridan tashkil topgan. Quyidagi amallar bajarilsin:
    - a) mualliflar familiyalari alfavit bo‘yicha tartiblangan ko‘rinishdagi kitoblar ro‘yxati chop etilsin;

- b) ko'rsatilgan nashriyot tomonidan chiqarilgan kitoblar ro'yxati chop etilsin;  
d) registratsiya nomeri ko'rsatilgan kitob fayldan o'chirilsin.
13. AVTO fayli <tabel nomeri>, <avtomobil egasi familiyasi>, <avtomobil nomeri> va <avtomobil turi> maydonlari bilan, avtomobillar texnik holatlari TA\_MIR faylida <tabel nomeri>, <avtomobilning texnik holati> maydonlari bilan berilgan. Avtomobilning texnik holati uchta holatda bo'lishi mumkin: 1-yaxshi; 2-qoniqarli; 3-qoniqarsiz. Quyidagi masalalar yechilsin:
- a) texnik holati yaxshi bo'lgan barcha avtomobillar haqidagi ma'lumot chop etilsin;  
b) ko'rsatilgan avtomobil nomeri bo'yicha uning texnik holati aniqlansin;  
d) texnik holati yaxshi bo'lgan barcha avtomobillar haqidagi ma'lumot fayldan o'chirilsin.
14. TELEFON fayli <telefon nomeri>, <abonent familiyasi>, <manzil> maydonlaridan iborat. Abonentning telefonda gaplashish vaqti GAPLASHUV faylida <telefon nomeri>, <gaplashuv vaqti> maydonlari bilan berilgan. Ayni vaqtdagi bir minutda gaplashish pul miqdori berilgan holat uchun quyidagilar amalga oshirilsin:
- a) ko'rsatilgan abonentga telefon uchun to'lov qog'ozi chop etilsin.  
b) gaplashuv vaqti 100 minutdan oshgan telefon nomerlari va ularning egalarining familiyalari chop etilsin.
15. TAOM faylida taom haqidagi ma'lumot <taom nomi>, <kaloriyasi> va <narxi> maydonlari bilan berilgan. KOMPLEKS faylida esa shu taomlardan tashkil qilingan taomlar majmualari <majmua nomeri>, <taom nomi> maydonlari orqali ko'rsatilgan. Har bir taom majmuasining tarkibi, umumiy kaloriyasi va narxi chop etilsin.
16. TEATR faylida <teatr kodi>, <teatr nomi> va <teatrdagi o'rinlar soni> maydonlari, BILET faylida <teatr kodi>, <spektakl nomi> va <sotilgan biletlar soni> maydonlari berilgan.  
Quyidagilar aniqlansin:
- a) bo'sh urinlari bor bo'lgan teatr va spektakl nomi;  
b) ko'rastilgan spektaklga bo'sh o'rin bor yoki yo'qligi;  
d) ko'rastilgan teatrdagi qanday spektakl bo'laytganligi.
17. Yo'lovchi umumiy yukini tavsiflovchi kattaliklar uning tarkibiga kiruvchi yuklar soni va umumiy og'irligi hisoblanadi.  
Berilgan YO\_LOVCHI faylda yo'lovchilarning yuklari haqidagi ma'lumot saqlanadi.  
Quyidagi masalalar yechilsin:
- a) shunday yuklar topilsinki, ularning har birining og'irligi barcha yuklar o'rtacha og'irligidan farqi 0.3 kg.dan oshmasin;  
b) yuklar soni bir xil, yuklar og'irligi bir biridan 0.5 kg.dan ko'p farq qilmaydigan ikkita yo'lovchi bor yoki yoqligi aniqlansin;  
d) yuklar soni, yuklar og'irligi bo'yicha boshqa yo'lovchilardan ortiq bolgan yo'lovchi bor yoki yoqligi aniqlansin;

- e) bitta yuki 30 kg.dan kam bo‘lmagan yo‘lovchi bor yoki yoqligi aniqlansin.
18. Eksport qilinuvchi tovarlar haqidagi ma’lumot - <tovar nomi>, <tavor export qilayotgan davlat nomi>, <tovar bahosi> va <tovar hajmi>, <tovar soni> EXPORT faylida berilgan. Ma’lum bir tovarni eksport qiluvchi davlat va shu tovarning umumiy eksportdagi hajmi aniqlansin.
  19. Butun sonlardan iborat binar **SERIYA** fayli berilgan. Undagi seriyalar chop etilsin (ya’ni, bir xil sonlardan iborat ketma-ketlik ostilari).
  20. Haqiqiy sonlardan iborat binar **LOKAL\_MM** fayli berilgan. Fayldagi lokal minimumlar va maximumlar miqdori aniqlansin. Lokal maksimum (minimum) deb sonlar ketma-ketligidagi  $a_{i-1} < a_i > a_{i+1}$  ( $a_{i-1} > a_i < a_{i+1}$ ) shartni qanoatlantiruvchi  $a_i$  soniga aytiladi.
  21. Haqiqiy sonlardan iborat **SONLAR** binar fayli berilgan. **SONLAR** faylining juft o‘rindagi (0,2,4,...) elementlaridan **JUFT\_SONLAR** fayli, toq o‘rindagi (1,3,5,...) elementlari **TOQ\_SONLAR** fayli hosil qilinsin.
  22. Haqiqiy sonlardan iborat binar **SON\_KVADRATI** fayli berilgan. Undagi barcha elementlar (sonlar) kvadratlari bilan almashtirilsin.
  23. Haqiqiy sonlardan iborat **YON\_ARIFM** binar fayli berilgan. Fayl boshidagi va oxiridagi elementlaridan boshqa har bir elementi, o‘zi va ikkita yon qo‘shnilarining o‘rta arifmetigi bilan almashtirilsin.
  24. Butun  $a_i, i = \overline{0, N-1}$  sonlardan iborat binary **MASSIV\_A** fayli berilgan ( $N > 0$ ). Fayldagi sonlarning boshlang‘ich joylashuvi quyidagicha o‘zgartirilsin:  
 $a_0, a_{h-1}, a_1, a_{h-2}, a_2, a_{h-3}, \dots$
  25. Ikkita butun  $i$  va  $j$  butun sonlar va matritsaning yuqori uchburchakli yoki pastki uchburchakli yoki uch diagonal elementlarini o‘zida satrlar bo‘yicha saqlagan haqiqiy sonlardan iborat **MATRITSA** fayl berilgan. Matritsa tartibini va  $i$ - satr va  $j$ - ustunda turgan elementni chop qiluvchi, agar bunday element topilmasa matritsaning nollaridan iborat bo‘lsa nolni chop qilsin aks holda -1 ni chop qilinsin.
  26. **A** va **B** matritsaning noldan farqli, yuqori uchburchak yoki pastki uchburchak yoki elementlarini satr bo‘yicha o‘zida saqlagan **MATR\_A**, **MATR\_B** fayllari berilgan.  
 Yangi **MATR\_C** fayl hosil qilinsin. Unda  $A*B$  elementlari satrlar bo‘yicha saqlansin, agar **A** va **B** matritsanini ko‘paytirish mumkin bo‘lmasa **MATR\_C** fayl bosh bo‘lsin.
  27.  $N$ ,  $M$  va  $S$  butun sonlari berilgan ( $0 < N, M < 10, -108 < S < 108$ ).  $N \times M$  o‘lchamdagi  $S = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} a_{ij}$  shartni qanoatlantiruvchi **A** matritsa hosil qilinsin.  $f$  binar faylga  $N$ ,  $M$  va **A** matritsa yozilsin.

## 22. Ko‘rsatkich turi

### *Namunaviy masala*

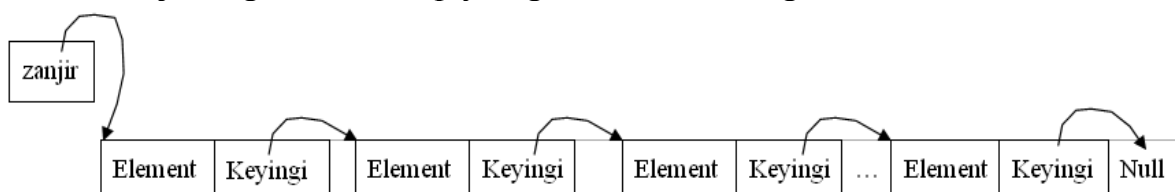
Noldan farqli butun sonlardan iborat chiziqli ro'yxat (zanjir) yaratilsin va undan ko'rsatilgan songa teng element o'chirilsin.

### ***Yechish usuli***

Butun sonlarning chiziqli ro'yxatning (zanjirning) xalqasi butun turdagi element va xuddi shunday zanjirga ko'rsatkich maydonlaridan tashkil topgan dinamik struktura ko'rinishida bo'ladi:

```
struct Zanjir
{
    int element;
    Zanjir * keyingi;
};
```

Zanjirning ko'rinishi quyidagi rasmda keltirilgan.



Programmaning bosh funksiyasida chiziqli ro'yxat hosil qilish uchun Zanjir turidagi zanjir o'zgaruvchisi aniqlanadi va unga bo'sh ko'rsatkich qiymati 0 beriladi (uning ekvivalenti - NULL). Keyin takrorlash operatori tanasida klaviaturadan butun son kiritiladi va zanjirga element joylashatirish funksiyasini chaqirish orqali bu son ro'yxatning oxiriga qo'shiladi. Funksiya yangi hosil bo'lgan ro'yxat boshining adresini yana zanjir o'zgaruvchisiga qaytaradi. Agar klaviaturadan 0 soni kiritilsa ro'yxatni hosil qilish jarayoni tugaydi. Ro'yxat ustida amal sifatida berilgan son bilan ustma-ust tushadigan elementlarni o'chirish uchun o'chiriladigan son o'zgaruvchiga o'qiladi va u zanjir elementini o'chirish funksiyasi chaqirilishida argument sifatida uzatiladi. Funksiya bu son bilan ustma-ust tushadigan ro'yxat elementlarini o'chiradi (agar bunday element mavjud bo'lsa) va o'zgargan ro'yxat boshining adresini zanjir o'zgaruvchisiga qaytarib beradi.

### ***Programma matni:***

```
#include <iostream.h>
struct Zanjir
{
    int element;
    Zanjir * keyingi;
};
Zanjir * Element_Joylash(Zanjir * z, int yangi_elem)
{
    Zanjir * yangi=new Zanjir;
    yangi->element=yangi_elem;
    yangi->keyingi=0;
```

```

if(z)                                // ro'yxat bo'sh emas
{
Zanjir * temp=z;
while(temp->keyingi)
    temp=temp->keyingi; // ro'yxat oxirgi elementini olish
temp->keyingi=yangi;      // yangi elementni ro'yxat oxiriga qo'shish
}
else z=yangi;              // ro'yxat bo'sh
return z;                  // ro'yxat bo'shi adresini qaytarish
}
Zanjir * Element_Uchirish(Zanjir * z, int del_elem)
{
if(z)
{
Zanjir* temp=z;
Zanjir* oldingi=0;        // joriy elementdan olingisiga ko'rsatgich
while (temp)
{
if (temp->element==del_elem)
{
if(oldingi)              // o'chiriladigan element birinchi emas
{
// o'chiriladigan elementdan oldingisini keyngisi
// bilan ulash
oldingi->keyingi = temp->keyingi;
delete temp;           // elementni o'chirish
temp=oldingi->keyingi;
}
else                      // o'chiriladigan element ro'yxat
boshida
{
z=z->keyingi;
delete temp;
temp=z;
}
}
else                      // element o'chiriladigan sondan farqli
{
oldingi=temp;
temp=temp->keyingi;
}
}
}
return z;

```

```

}
void Zanjir_Ekranga(Zanjir * z)
{
    cout<<"Zanjir elementlari:"<<endl;
    Zanjir * temp=z;
    while(temp)
    {
        cout<<temp->element<<' ';
        temp=temp->keyingi;
    }
    cout<<endl;
}
Zanjir * Zanjirni_Uchirish(Zanjir * z)
{
    Zanjir * temp=z;
    while(z)
    {
        z=z->keyingi;
        delete temp;
    }
    return z;
}
int main()
{
    Zanjir * zanjir=0;
    int son,del_element;
    do
    {
        cout<<"\nSonni kiriting (0-jaryon tugatish): ";
        cin>>son;
        if(son) zanjir=Element_Joylash(zanjir,son);
    }
    while(son);
    Zanjir_Ekranga(zanjir);
    cout<<"\nO'chiriladigan elementni kiriting: ";
    cin>>del_element;
    zanjir= Element_Uchirish(zanjir,del_element);
    Zanjir_Ekranga(zanjir);
    Zanjir = Zanjirni_Uchirish(zanjir);
    return 0;
}

```

### ***Amaliy topshiriqlar***

1. Z zanjir int turidagi elementga (element maydoni) ega. Z zanjir elementlarining o'rta arifmetigi topilsin.

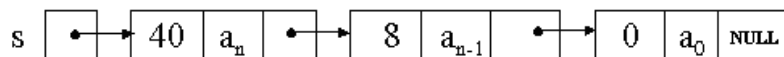
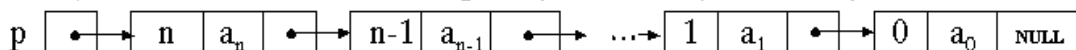
2.  $Z$  zanjir elementi satr turida.  $Z$  zanjirga kiruvchi barcha  $Suz1$  qiymatiga teng elementlar  $Suz2$  bilan almashtirilsin.
3.  $Z$  zanjir elementi satr turida.  $Z$  zanjirdagi bir xil belgi bilan boshlanuvchi va tugaydigan elementlar soni aniqlansin.
4.  $Z$  zanjiri elementi satr turida.  $Z$  zanjirning oxirgi elementi bilan ustma-ust tushadigan elementlar soni aniqlansin.
5.  $Z$  zanjir elementlari butun turda.  $Z$  zanjirdan ikkita  $Z_1$  va  $Z_2$  zanjirlar hosil qilinsin. Bunda  $Z_1$  - zanjirning musbat elementlari va  $Z_2$  - qolgan elementlari.
6. Oraga qo'yishni bajaruvchi funksiyalar tuzilsin:
  - a)  $Z$  zanjir boshiga yangi Yangi elementni qo'yadigan;
  - b)  $Z$  zanjir oxiriga yangi Yangi elementni qo'yadigan.
7. Oraga qo'yishni bajaruvchi funksiyalar tuzilsin:
  - a) bo'sh bo'lmagan  $Z$  zanjirining birinchi elementidan keyin yangi  $y$  elementni;
  - b)  $Z$  zanjiriga kiruvchi har bir  $y$  elementdan keyin yangi  $x$  elementni joylashtiradigan.
8. Bo'sh bo'lmagan va tartiblangan  $Z$  zanjirga yangi  $y$  elementni shunday joylashtiringki, unda  $Z$  zanjirdagi tartib buzilmasin.
9. Zanjirning elementlarini o'chiradigan funksiya tuzilsin:
  - a) bo'sh bo'lmagan  $Z$  zanjirning birinchi elementini;
  - b)  $Z$  zanjirning ikkinchi elementini, agar u mavjud bo'lsa.
10. Butun turdagi elementlardan iborat  $Z$  zanjir elementini o'chiradigan funksiya tuzilsin:
  - a) bo'sh bo'lmagan  $Z$  zanjirning oxirgi elementini;
  - b)  $Z$  zanjirdan birinchi manfiy element, agar u mavjud bo'lsa;
  - d)  $Z$  zanjirdagi barcha manfiy elementlarni.
11. Klaviaturadan matnni (satrni) o'qib, teskari tartibda chop qilinsin. Bunda matndagi har bir so'z zanjir elementi sifatida qaralsin.
12. Butun  $n$  ( $n > 1$ ) va  $n$  ta haqiqiy sonlar berilgan. Bu sonlar kamaymaydigan tartibda chop qilinsin. Bunda har bir son zanjir elementi sifatida qaralsin.
13. Quyidagi amallarni bajaruvchi funksiya tuzilsin:
  - a)  $Z_1$  va  $Z_2$  zanjirlar tengligini tekshiruvchi;
  - b)  $Z_1$  zanjir  $Z_2$  zanjirga kirishini aniqlovchi.
14. Quyidagi amallarni bajaruvchi funksiya tuzilsin:
  - a)  $Z$  zanjirda kamida ikkita bir xil element bor-yo'qligini aniqlovchi;
  - b) bo'sh bo'lmagan  $Z$  zanjir oxiriga uning birinchi elementini olib o'tuvchi;
  - d) bo'sh bo'lmagan  $Z$  zanjir boshiga uning oxirgi elementini olib o'tuvchi.
15. Quyidagi amallarni bajaruvchi funksiya tuzilsin:
  - a)  $Z_1$  zanjir oxiriga  $Z_2$  zanjirining barcha elementlarini qo'shuvchi;
  - b) agar  $Z_1$  zanjirda  $x$  element mavjud bo'lsa, uning davomiga  $Z_2$  zanjirning barcha elementlarini qo'shuvchi.
16. Quyidagi amallarni bajaruvchi funksiya tuzilsin:



- a)  $Z$  zanjirda ketma-ket keluvchi teng qiymatli elementlar guruhidan bittasini qoldiruvchi;
- b)  $Z$  zanjirda bir xil qiymatli elementlardan faqat bittasini qoldiruvchi.
17. Berilgan shartlarni bajaruvchi rekursiv funksiya tuzilsin:
- a)  $e$  element  $Z$  zanjirga kirishi yoki kirmasligini aniqlovchi;
- b)  $e$  element  $Z$  zanjirga necha marta kirishini hisoblovchi.
18. Quyidagi shartlarni bajaruvchi rekursiv funksiya tuzilsin:
- a) bo'sh bo'lmagan, haqiqiy turdagi elementli  $Z$  zanjirga kiruvchi maksimal qiymatli elementni topuvchi;
- b)  $Z$  zanjir elementlarini teskari tartibda chop qiluvchi.
19. Quyidagi shartlarni bajaruvchi rekursiv funksiya tuzilsin:
- a)  $Z$  zanjirdagi barcha  $e_1$  elementni  $e_2$  elementga almashtiruvchi;
- b)  $Z$  zanjirdan qiymati  $e$  ga teng bo'lgan 1-elementning kirishini o'chiruvchi (agar u mavjud bo'lsa).
20. Quyidagi shartlarni bajaruvchi rekursiv funksiya tuzilsin:
- a)  $Z$  zanjirdan berilgan  $e$  elementning barchasini o'chiruvchi;
- b)  $Z$  zanjirining nusxasi –  $Z_1$  zanjirni hosil qiluvchi.
21. Berilgan  $Z_1$  va  $Z_2$  zanjirlar bo'yicha quyidagi shartlarni qanoatlantiruvchi elementlardan tashkil topgan  $Z$  zanjir hosil qilinsin:
- a)  $Z_1$  va  $Z_2$  zanjirlarining kamida bittasiga kiruvchi;
- b) bir vaqtda  $Z_1$  va  $Z_2$  zanjirlariga kiruvchi;
- d)  $Z_1$  zanjirga kiruvchi, lekin  $Z_2$  zanjirga kirmaydigan;
- e)  $Z_1$  va  $Z_2$  zanjirlardan bittasiga kiruvchi, lekin ikkinchisiga kirmaydigan.
22. Kamayuvchi bo'lmagan  $Z_1$  va  $Z_2$  zanjirlarni kamayuvchi bo'lmagan zanjirga birlashtiruvchi funksiya tuzilsin:
- a) yangi  $Z$  zanjir qurish orqali;
- b)  $Z_1$  va  $Z_2$  zanjirdagi ko'rsatgichlarni mos ravishda o'zgartirish va natijani  $Z_1$  zanjirida hosil qilish orqali.
23.  $Z$  zanjiridagi  $Z_1$  zanjir ostining birinchi kirishini  $Z_2$  zanjir bilan almashtiruvchi funksiya tuzilsin.
24. Butun koeffitsiyentli

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + a_1 x + a_0$$

ko'phadni zanjir ko'rinishida tasvirlash mumkin (quyidagi rasmdagi  $p$  zanjir), agar  $a = 0$  bo'lsa, mos xalqa zanjirga kiritilmaydi. Pastdagi rasmda  $s$  zanjir  $S(x) = 52x^{40} - 3x^8 + x$  ko'phadga mos zanjir keltirilgan.



Ko'phadni tavsiflashning zanjir ko'rinishiga mos turlar e'lon qilinsin va bu zanjir ustida quyidagi amallarni bajaruvchi funksiyalar tuzilsin:

- a)  $p$  va  $q$  ko'phadlarni tengligini tekshiruvchi  $Teng(p,q)$  mantiqiy funksiyasi;
  - b) berilgan  $x$  uchun  $p$  ko'phadning qiymatini hisoblovchi  $Qiymat(p,x)$  funksiyasi;
  - d)  $p$  ko'phadning hosilasi bo'lgan  $q$  ko'phadni quruvchi  $Hosila(p,q)$  funksiyasi;
  - e)  $q$  va  $r$  ko'phadlar yig'indisi bo'lgan  $p$  ko'phadni quruvchi  $Yig'indi(p,q,r)$  funksiyasi;
  - f)  $p$  ko'phadni  $v$  o'zgaruvchining qiymati bo'lgan (bitta harfli) o'zgaruvchi nomida chop qiluvchi  $Chop\_Qilish(p,v)$  funksiyasi tuzilsin. Misol uchun,  $S$ -ko'phad uchun  $Chop\_Qilish(S,'y')$  funksiyasi " $52y^{40}-3y^8+y$ " ifodasini chop qiladi.
25. ("*Sanagich*")  $n$  ta bola aylana bo'ylab turibdi. Sanoq birinchisidan boshlanib,  $k$ -bola davradan chiqariladi va har bir chiqarishdan keyin davra qisqaradi. Keyingi sanoq  $k+1$ -boladan boshlanadi. Bolalarni davradan chiqib ketish tartibi aniqlansin. Programma uchun boshlang'ich berilganlar  $n$  va  $k$  natural sonlar bo'lib, uning natijasi – davradan chiqib ketuvchi bolalarning boshlang'ich tartib nomerlari ketma-ketligi.
26. Berilgan matn (satr) simmetrik ekanligi aniqlansin. So'zlar zanjir elementi deb qaralsin.
27. Kamida ikkita har xil va  $0$  bilan tugaydigan natural sonlar ketma-ketligi berilgan. Eng katta va eng kichik sonlar o'rtasidagi sonlar teskari tartibda chop qilinsin. Sonlar zanjir elementi deb qaralsin.

### 23. Sinflar. Inkapsulyatsiya

Quyida masalalarni yechishda sinf yaratilishi va unda qo'yilgan masalani to'liq qamrab oluvchi berilganlar-a'zolar va funksiya-a'zolar aniqlanishi zarur.

#### ***Namunaviy masala***

Berilgan matn Shenon usuli bilan shifrlansin va qayta tiklansin.

#### ***Yechish usuli***

Shenon usulining mohiyatida  $a$  baytni  $b$  baytga "*inkor qiluvchi yoki*" amalini ketma-ket ravishda ikki marta qo'llash natijasida yana  $a$  baytning o'zi tiklanishi yotadi:  $a^b b^a = a$ .

Shifrlanuvchi matn ( $Matn_0$ ) satrlar ko'rinishida, shifrlash kaliti ( $Kalit$ ) esa bayt ko'rishida beriladi.  $Matn_0$  matnning satrlaridagi har bir bayt bilan  $Kalit$  o'rtasida  $\wedge$  amalini bajarish orqali shifrlangan  $Matn_1$  matni hosil qilinadi. Xuddi shu amalni  $Matn_1$  matniga qo'llash orqali  $Matn_0$  matni qayta tiklanadi.

Masalani yechish algoritmi quyidagi qadamlardan iborat bo'ladi:

1.  $Matn_0$  satrlarini kiritiladi;
2.  $Kalit$  qiymatini tasoddiy ravishda hosil qilinadi;
3.  $Matn_0$  matnning har bir satrini  $Kalit$  bilan shifrlash asosida  $Matn_1$  satrlarini hosil qilinadi.

4. Matn1 matnning har bir satrini Kalit bilan “qayta shifrlash” asosida Matn2 matni hosil qilinadi.

***Programma matni***

```
TForm *Form1;
class Shifr_Shenon
{
    String matn0,matn1,matn2, kalit;
    unsigned char kalit_son;
    String _10dan_2ga(unsigned char c);
    String matn_bin_matn_char(String mtn);
public:
    void Matnni_kiritish(String _S);
    void Shifrlash();
    void Shifrni_ochish();
    String Matndan_01ga(String);
    String _01dan_Matnga();
    char _01dan_Belgiga(String);
    String Kalitni_olish(){return kalit;};
    char Kalit_belgini_olish(){return kalit_son;};
    String Matn0_olish(){return matn0;};
    String Matn1_olish(){return matn1;};
    String Matn_01dan_Matn_char(int);
};
String Shifr_Shenon::_10dan_2ga(unsigned char c)
{
    String Dicemal="";
    for (int k=1; k<=8;k++)
    {
        if(c&0x80) Dicemal=Dicemal+'1';
        else Dicemal=Dicemal+'0';
        c=c<<1;
    }
    return Dicemal;
};
void Shifr_Shenon::Matnni_kiritish(String _S)
{
    matn0=Matndan_01ga(_S);
    unsigned int kod;
    randomize();
    kod=rand();
    kalit_son=kod%256;
    kalit=_10dan_2ga(kalit_son);
    while(kalit[1]=='0') kalit.Delete(1,1);
};
```

```

void Shifr_Shenon::Shifrlash()
{
    matn1.operator =("");
    int len_kalit=kalit.Length();
    for(int i=0;i<matn0.Length();i++)
    {
        if(matn0[i+1]=='0'&&kalit[i%len_kalit+1]=='0'
        || matn0[i+1]=='1'&&kalit[i%len_kalit+1]=='1')
            matn1+="0";
        else matn1+="1";
    }
};

void Shifr_Shenon::Shifrni_ochish()
{
    matn2.operator =("");
    int len_kalit=kalit.Length();
    for(int i=0; i<matn1.Length();i++)
    {
        if(matn1[i+1]=='0'&&kalit[i%len_kalit+1]=='0'
        || matn1[i+1]=='1'&&kalit[i%len_kalit+1]=='1')
            matn2+="0";
        else matn2+="1";
    }
};

String Shifr_Shenon::Matndan_01ga(String matn)
{
    String sbin="";
    for(int i=1; i<=matn.Length(); i++)
    {
        unsigned char c=(unsigned char)matn[i];
        sbin.operator+=(_10dan_2ga(c));
    }
    return sbin;
};

char Shifr_Shenon::_01dan_Belgiga(String sbin)
{
    unsigned int koda=0, pow2=1;
    for(int i=8;i>=1;i--)
    {
        koda+=((unsigned char)(sbin[i])-48)*pow2;
        pow2=pow2*2;
    }
    return koda;
}

```

```

String Shifr_Shenon::matn_bin_matn_char(String mtn)
{
    String matn="";
    while (mtn.Length())
    {
        matn+=_01dan_Belgiga(mtn.SubString(1,8));
        mtn.Delete(1,8);
    }
    return matn;
};

String Shifr_Shenon::Matn_01dan_Matn_char(int imatn)
{
    if(imatn==1) return matn_bin_matn_char(matn1);
    return matn_bin_matn_char(matn2);
};

Shifr_Shenon shenon;

void __fastcall TForm::Button_shifrlashClick(TObject *Sender)
{
    Button_shifrlash->Enabled=0;
    shenon.Shifrlash();
    Memo_matn1_01->Lines->Text=shenon.Matn1_olish();
    Memo_matn1->Lines->Text=shenon.Matn_01dan_Matn_char(1);
    Button_shifrni_ochish->Enabled=1;
}

void __fastcall TForm::Button_shifrni_ochishClick(TObject *Sender)
{
    shenon.Shifrni_ochish();
    Memo_matn2->Lines->Text=shenon.Matn_01dan_Matn_char(2);
}

void __fastcall TForm::Edit_kalitChange(TObject *Sender)
{
    if (Memo_matn0_01->Lines->Text!=""
        &&Panel_Kalit->Caption!="")
        Button_shifrlash->Enabled =1;
    else Button_shifrlash->Enabled =0;
}

void __fastcall TForm::Memo_matn0Change(TObject *Sender)
{
    if (Memo_matn0->Lines->Text!="")
        Button_01_kod->Enabled =1;
    else Button_01_kod->Enabled =0;
    Memo_matn0_01->Lines->Clear();
}

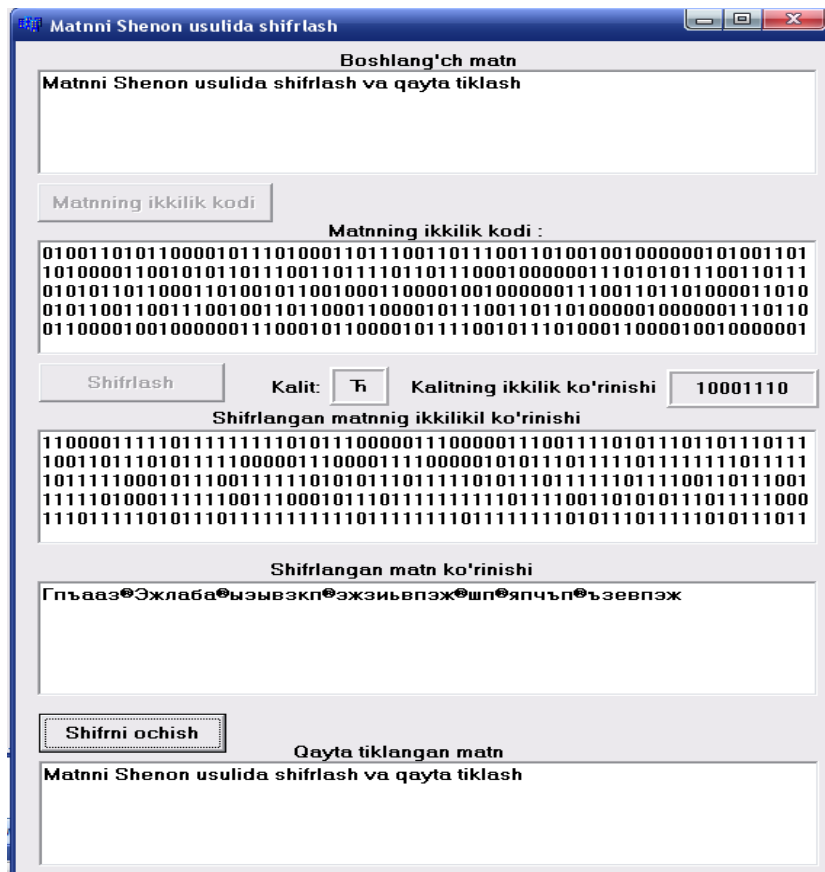
```

```

Memo_matn1_01->Lines->Clear();
Memo_matn2->Lines->Clear();
}
void __fastcall TForm::Button_01_kodClick(TObject *Sender)
{
shenon.Matnni_kiritish(Memo_matn0->Lines->Text);
Memo_matn0_01->Lines->Text=shenon.Matn0_olish();
Panel_Kalit_Bin->Caption=shenon.Kalitni_olish();
Panel_Kalit->Caption=shenon.Kalit_belgini_olish();
Button_shifrlash->Enabled=1;
Button_01_kod->Enabled=0;
}
void __fastcall TForm::FormCreate(TObject *Sender)
{
Button_01_kod->Enabled=0;
Button_shifrlash->Enabled=0;
Button_shifrni_ochish->Enabled=0;
Memo_matn0_01->ReadOnly=1;
Memo_matn1_01->ReadOnly=1;
Memo_matn2->ReadOnly=1;
}

```

Ilovaning bajarilshi quyidagi rasmda ko‘rsatilgan.



### *Amaliy topshiriqlar*

1. 10 lik sanoq sistemasida berilgan sonni 2, 8 va 16 lik sanoq sistemasidagi ko‘rinishini chop qiluvchi **SANOQ\_SISTEMA** sinfi yaratilsin.
2. Kompleks sonlar ustida arifmetik amallarni bajaruvchi **KOMPLEKS** sinfi yaratilsin.
3. Berilgan natural  $n$  soni uchun  $n \times n$  o‘lchamidagi **A** matritsani maksimal va minimal elementini topadigan, uning bosh diagonalga nisbatan simmetrik ekanligini aniqlaydigan, transponerlangan ko‘rinishini chop etuvchi funksiya-a‘zolarini o‘z ichiga oluvchi **MATRITSA** sinfi yaratilsin.
4. Uch o‘lchamli fazoda koordinatalari bilan berilgan ikkita vektorni tavsiflovchi **VEKTOR2\_3D** sinfi aniqlansin. Sinfda vektorlarni qo‘shish va ayirish orqali yangi vektorlar hosil qiluvchi, ikkita vektorning skalyar ko‘paytmasini, vektor uzunligini va ikkita vektorlar orasidagi burchak kosinusini hisoblovchi funksiya-a‘zolar aniqlansin.
5. Ko‘phad darajasi va koeffisientlari bilan berilgan bitta o‘zgauvchili ko‘phadni tavsiflovchi **KO\_PHAD** sinfi yaratilsin. Sinfda ko‘phad berilgan argumentdagi qiymatini hisoblovchi, ko‘rsatilgan tartibdagi hosilasini topadigan funksiya-a‘zolar aniqlansin.
6. Uy kutubxonasini tavsiflovchi **UY\_KUTUBXONASI** sinfi aniqlansin. Unda ixtiyoriy sondagi kitoblar bilan ishlash, qandaydir alomati boyocha kitobni izlash (muallif yoki yil bo‘yicha), yangi kitobni qo‘shish va o‘chirish imkoniyatlari bo‘lsin.
7. Yon daftarni o‘zida aks ettiruvchi **YON\_DAFTAR** sinfi yaratilsin. Unda ixtiyoriy sondagi yozuvlar bilan ishlash, qandaydir alomati boyocha yozuvni izlash (familiya, tug‘ilgan yili yoki telefon nomeri bo‘yicha), yangi yozuvni qo‘shish va o‘chirish imkoniyatlari bo‘lsin.
8. Talabalar guruhini tavsiflovchi **TALABA\_GURUHI** sinfi yaratilsin. Unda ixtiyoriy sondagi talabalar bilan ishlash, qandaydir alomati boyocha talabani izlash (familiya, tug‘ilgan yili yoki telefon nomeri bo‘yicha), yangi yozuvni qo‘shish, o‘chirish va tartiblash imkoniyatlari bo‘lsin.
9. Hayvonlarning o‘zaro raqobat qiluvchi ikkita turining  $n$  - yildagi bir-birining  $x_n$  va  $y_n$  o‘lchamlariga (sonlariga) o‘zaro ta’siri quyidagi sistema bilan tavsiflanadi:

$$x_{n+1} = 2x_n - y_n,$$

$$y_{n+1} = -x_n + 2y_n.$$

Boshlang‘ich yildagi sonlari -  $x_0$  va  $y_0$  berilganda birorta turning to‘la qirilib ketguncha bo‘lgan vaqt oralig‘idagi turlar sonidagi o‘zgarishlar dinamikasini chop etuvchi **POPULYATSIYA** sinfi aniqlansin.

10. Stek ustidagi amallarni bajaruvchi **STEK** sinfi aniqlansin. Ushbu sinfdan labirintdan chiqish masalasini yechishda foydalanilsin. Labirint kvadratlardan tashkil topgan matritsa ko‘rinishida beriladi. Har bir kvadrat ochiq yoki yopiq bo‘ladi. Yopiq kvadratga kirish mumkin emas. Agar kvadrat ochiq bo‘lsa uning yon tomonidan kirish mumkin (burchagidan

kirish mumkin emas). Har bir kvadrat uning matritsadagi koordinatalari bilan beriladi. Labirintdan chiqich amalga oshirilganda topilgan yo‘l chop qilinadi (kvadratlar koordinatalari juftliklarining ketma-ketligi).

11. YUGURUVCHI sinfi yugurish musobaqasi natijalari haqidagi <yuguruvchi familiyasi va initsiallari>,<jamoa nomi> va <masofani bosib o‘tgan vaqti (sekundlarda)> berilgan-a’zolari o‘z ichiga oladi. Berilgan n o‘lchamidagi YUGURUVCHI sinf obe’ktlari massivi yaratilsin va quyidagi amallar bajarilsin:
  - a) massiv yuguruvchilarni masofani bosib o‘tgan vaqtining kamayishi bo‘yicha tartiblansin;
  - b) jamoa a’zolarining o‘rtacha yugurish vaqti bo‘yicha yuqori natija ko‘rsatgan uchta jamoa nomlari chop etilsin.
12. FUTBOL sinfi jamoasining o‘yin natijalari haqidagi <jamoa nomi>,<g‘alabalar soni>,<duranglar soni>,<mag‘lubiyatlar soni>,<kiritgan to‘plar soni>va <o‘tkaz-gan to‘plar soni> berilgan-a’zolari o‘z ichiga oladi.  
Berilgan n uchun FUTBOL sinfi obyektlari massivi hosil qilinsin va to‘plagan ochkolari bo‘yicha jamolar jadvali chop etilsin. Bunda quyidagilarga e’tibor berilsin: agar ikkita jamoaning ochkolari teng bo‘lsa, kiritilgan va o‘tkazib yuborilgan to‘plar farqi qaraladi. Farqi katta bo‘lgan jamoa uyqori qatorga o‘tadi, aks holda qur’a tashlanadi va shunga qarab jamoa o‘rni aniqlanadi.
13. AVTOMOBIL sinfida avtomobil va uning egasi haqidagi <avtomobil egasining familiyasi va initsiallari>,<avtomobil rusumi>,<avtomobil nomeri>berilgan-a’zolari aniqlangan. Berilgan n o‘lchamidagi AVTOMOBIL sinfi obyektlari massivi yaratilsin va quyidagi amallar bajarilsin:
  - a) massiv avtomobil egalarining familiyalarini alfavit bo‘yicha joylashuviga mos tartiblansin;
  - b) kiritilgan avtomobil rusumidagi avtomobil egalari haqidagi ma’lumot chop etilsin;
  - d) kiritilgan avtomobil rusumi va nomeri bo‘yicha avtomobil egasining familiyasi chop etilsin.
14. Nuqta bilan tugaydigan matn berilgan. Uning formula ekanligi quyidagi qoidalar asosida tekshirilsin. Natija ijobiy bo‘lgan holda formula qiymati hisoblansin:
 
$$\langle \text{formula} \rangle ::= \langle \text{raqam} \rangle | (\langle \text{formula} \rangle \langle \text{belgi} \rangle \langle \text{formula} \rangle)$$

$$\langle \text{belgi} \rangle ::= + | - | *$$

$$\langle \text{raqam} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$$
 Masalan, “5” formula qiymati 5, ”((2–4)\*6)” formula qiymati 12.
16. TO\_RTBURCHAK sinfi yaratilsin. Uning tarkibida to‘rtburchak tomonlari- a,b,c,d kiritilganda uning mavjudligini, agar mavjud bo‘lsa to‘rburchakning yuzasini, perimetri va turini (to‘g‘riburchakli, kvadrat, parallellogram) aniqlovchi funksiya-a’zolar tuzilsin.



17. Butun sonlar juftligi bilan berilgan ratsional sonlar ustida amal bajaruvchi **RATSIONAL** sinfi aniqlansin. Sinf obyektlari massivini yaratilsin va sinfnings do'st funksiyalari yordamida quyidagi masalalar yechilsin:  
 a) berilgan  $a$  va  $b$  ratsional sonlarning tengligi tekshirilsin;  
 b) berilgan  $a$  va  $b$  ratsional sonlar yig'indisi  $r$  ratsionalga berilsin;  
 d) berilgan  $r$  ratsional sonni qisqartirib bo'lmaydigan ko'rinishga keltirilsin;  
 massiv ko'rinishida berilgan ratsional sonlar ichida eng kattasi topilsin.
18. O'zaro ekvivalent formulardan foydalangan holda berilgan mantiqiy ifodani soddalashtirish amalini bajaradigan **MANTIQUIY\_IFODA** sinfi yaratilsin. Quyida o'zaro ekvivalent formulalar ro'yxati keltirilgan.
- 1)  $1 \wedge x \equiv x$ ;                      2)  $x \vee 1 \equiv 1$ ;                      3)  $x \wedge 1 \equiv x$ ;  
 4)  $x \vee 0 \equiv x$ ;                      5)  $x \vee (x \wedge y) \equiv x$ ;                      6)  $x \wedge (x \vee y) \equiv x$ ;  
 7)  $x \wedge x \equiv x$ ;                      8)  $x \vee x \equiv x$ ;                      9)  $x \wedge 1 \equiv 0$ ;  
 10)  $1(x \vee y) \equiv 1x \wedge 1y$ ;    11)  $1(x \wedge y) \equiv 1x \vee 1y$ ;    12)  $x \wedge y \equiv y \wedge x$ ;  
 13)  $x \vee y \equiv y \vee x$ ;              14)  $(x \wedge y) \wedge z \equiv x \wedge (y \wedge z)$ ;    15)  $(x \vee y) \vee z \equiv x \vee (y \vee z)$ ;  
 16)  $x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$ ;    17)  $x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$ ;    18)  $(x) \equiv x$ .

## 24. Vorislik

Ushbu bo'limda qo'yilgan masala mazmunidan kelib chiqqan masalaning umumiy xususiyatlarini o'z ichiga olgan tayanch sinf yaratilishi va undan voris sifatida hosil bo'lgan sinf esa bevosita qo'yilgan masalani yechishi kerak. Masala bosh funksiyada yaratilgan voris sinf ob'jektini e'lon qilib echiladi.

### *Namunaviy masala*

Berilgan butun  $n$ , va haqiqiy  $x$  va  $y$  qiymatlari uchun  $10^n - x^n + x^n$  ifoda hisoblansin.

### *Yechish usuli*

Berilgan ifoda qiymati uchta ifoda ostilari qiymatlarini alohida hisoblash va mos arifmetik amallar bajarish ko'rinishida bajariladi. Har bir ifoda o'zaro vorislik munosabatida bo'lgan uchta sinf obyektlari sifatida qaraladi.

Tayanch sinf sifatida  $x^n$  ifodani hisoblash uchun yaratilgan `lfoda_x_n` sinfi bo'lib, uning tarkibiga butun  $n$ , haqiqiy  $x$  berilgan-a'zolari, berilgan-a'zolarga qiymat beruvchi konstruktor, nusxalash konstruktori va bevosita  $x^n$  qiymatini hisoblaydigan `Hisobla()` funksiya-a'zosi kiradi.

Tayanch `lfoda_x_n` vorisi sifatida  $x^n$  ifoda osti uchun `lfoda_x_y` sinfi yaratilgan. Unda haqiqiy  $y$  berilgan-a'zo aniqlangan bo'lib,  $x$  qiymati tayanch sinfdan voris sifatida olinadi. Ifoda ostilari o'zaro bog'liqligini inobatga oladigan bo'lsak, `lfoda_x_y` sinfi obyekti `lfoda_x_n` obyekt mavjud bo'lgandagina yuzaga kelishi mumkin va shu sababli unda nusxalash

konstruktorlari qo'llanilgan. Hisobla() funksiyasi  $x^n$  ifoda ostisi qiymatini hisoblaydi.

O'z navbatida  $10^n$  ifoda ostisi uchun Ifoda\_10\_y sinfi Ifoda\_x\_y sinfi vorisi qilib aniqlangan. Unda berilgan-a'zolar yo'q, y qiymati ajdod sinflardan vorislik bo'yicha olinadi. Hisobla() funksiya-a'zosi  $10^n$  qiymatini hisoblaydi.

Qo'yilgan masala mazmuniga ko'ra sinflar obyektlarini aniqlash qat'iy ketma-katlikda amalga oshiriladi. Oldin Ifoda\_x\_n, keyin unga bog'liq Ifoda\_x\_y obyekt va nihoyat Ifoda\_10\_y sinf obyektlari aniqlanadi va umumiy ifoda qiymati hisoblanadi.

### ***Programma matni***

#### ***Unit\_Ifoda.h fayli:***

```
class Ifoda_x_n
{
protected:
double x;
int n;
public:
Ifoda_x_n(int _n, double _x){ n=_n; x=_x;}
Ifoda_x_n(Ifoda_x_n & if1){ n=if1.n; x=if1.x; }
double Hisobla(){ return pow(x,n);}
};
class Ifoda_x_y:public Ifoda_x_n
{
protected:
double y;
public:
Ifoda_x_y(double _y,Ifoda_x_n & if1):Ifoda_x_n(if1)
{ y=_y; }
Ifoda_x_y(Ifoda_x_y & if12): Ifoda_x_n(if12.n,if12.x)
{ y=if12.y;}
double Hisobla(){ return pow(x,y);}
};
class Ifoda_10_y:public Ifoda_x_y
{
public:
Ifoda_10_y(Ifoda_x_y& if_1_2):Ifoda_x_y(if_1_2){}
double Hisobla(){return pow(10,y);}
};
Ifoda_x_n * if_xn;
Ifoda_x_y* if_xy;
Ifoda_10_y * if_10y;
```

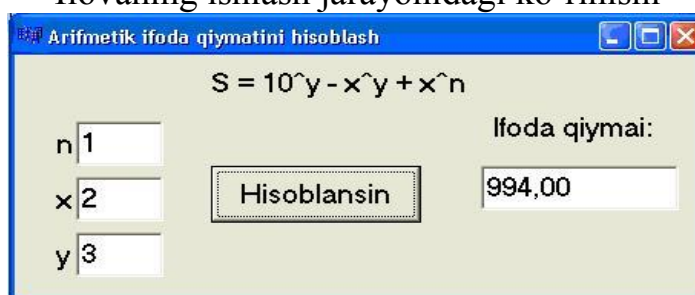
#### ***Unit\_Ifoda.cpp fayli:***

```

void __fastcall TForm1::Btn_HisoblaClick(TObject *Sender)
{
if_xn=new Ifoda_x_n(StrToInt(Edit_n->Text),StrToFloat(Edit_x->Text));
if_xy=new Ifoda_x_y(StrToFloat(Edit_y->Text),*if_xn);
if_10y=new Ifoda_10_y(*if_xy);
Edit_S->Text=FloatToStrF(if_xn->Hisobla()-if_xy->Hisobla()      +if_10y-
>Hisobla(),2,6,2);
delete if_xn;
delete if_xy;
delete if_10y;
}

```

Ilovaning ishlash jarayonidagi ko‘rinishi



### *Amaliy topshiriqlar*

1. 10 lik sanoq sistemasida berilgan ishorasiz haqiqiy sonni ko‘rsatilgan “p.m” formatda chop qiladigan FLOAT\_PRINT sinf yaratilsin. Bu yerda p-son ko‘rinishidagi jami joylar (pozitsiya), m-sonning kasr qismidagi raqamlar soni. Natijani 2,8 va 16 lik sanoq sistemasidagi ko‘rinishini chop qiluvchi FLOAT\_PRINT2, FLOAT\_PRINT8, FLOAT\_PRINT16 hosilaviy sinflar yaratilsin.
2. Kompleks sonlar ustida arifmetik amallar bajaradigan KOMPLEKS tayanch sinfi yaratilsin. Undan voris sinf sifatida kompleks koeffisientli kvadrat tenglama ildizini topadigan KOMP\_KV\_TENGLAMA sinfi yaratilsin.
3. Berilgan natural n o‘lchamidagi  $a_{ij} (1 \leq i, j \leq n)$  haqiqiy elementli kvadrat matritsa uchun xotiradan joy ajratich, qiymatlarini o‘qish va chop qilish amallarini bajaradigan MATRITSA tayanch sinfi yaratilsin. Uning vorisi qilib matritsaning determinantini minorlar usulida hisoblovchi funksiya-a’zosi bo‘lgan DTR\_MATRITSA sinfi yaratilsin.

Matritsa determentanti quyidagicha hisoblanadi:

$$\Delta_n = \sum_{k=1}^n a_{1k} A_{1k}.$$

Bu yerda  $A_{1k} = (-1)^{1+k} M_{1k}$ ,  $M_{1k}$  - minor bo‘lib, u  $a_{ij}$  matritsaning 1-satri va k-ustunini o‘chirishdan hosil bo‘ladigan (n-1) - tartibli matritsaning determenanti.

4. n o‘lchamli fazoda koordinatalari bilan berilgan ikkita vektorni tavsiflovchi,

ya'ni ularni xotirada saqlash, qiymatlarini o'qish va chop qilish amalini bajaruvchi **VEKTOR\_XY** tayanch sinfi aniqlansin. Uning vorisi bo'lgan **VEKTOR\_XY\_AMAL** sinfida vektorlarni qo'shish va ayirish orqali yangi vektorlar hosil qiluvchi, ikkita vektorning skalyar ko'paytmasini, vektor uzunligini va ikkita vektorlar orasidagi burchak kosinusi hisoblovchi funksiya-a'zolar aniqlansin.

5. Ko'phad darajasi va koeffisientlari bilan berilgan bitta o'zgavchili ko'phadni xotirada saqlash, qiymat o'qish va chop qilish amalini bajaruvchi **KOP\_HAD** tayanch sinfi yaratilsin. Ushbu sinf vorisi sifatida berilgan butun  $k$  soni uchun  $k$ -tartibli Chebishev ko'phadi koeffisientlarini hisoblaydigan va berilgan haqiqiy turdagi argumenti uchun ko'phad qiymatini hisoblovchi **SHEBISHEV** sinfi yaratilsin (13.12-masalaga qaralsin).
6. Kitobning nomi, muallifi, nashriyoti nomi va chop qilingan yili bo'yicha berilganlarni xotirada saqlash, qiymat o'qish va chop qilish amalini bajaruvchi **KITOB** tayanch sinfi yaratilsin. Uning vorisi bo'lgan **UY\_KUTUBXONASI** sinfida - uy manzili, kutubxona egasi familiya, ismi haqida ma'lumotlar bo'lsin.  
**UY\_KUTUBXONASI** sinfida obyektlari chiziqli zanjirini yaratgan holda unda ixtiyoriy sondagi kitoblar bilan ishlash, qandaydir alomati bo'yicha kitobni izlash (muallif, yil yoki uy manzil bo'yicha va hokazo), yangi kitobni qo'shish va o'chirish amalga oshirilsin.
7. Berilgan satrni saqlash va chop qilish amallarini o'z ichiga olgan **SATR** sinfi aniqlansin. **SATR** sinfining vorisi sifatida **ARIFM\_AMAL** sinfi yaratilsinki, unda satr ko'rinishida berilgan ikkita son o'rtasida arifmetik amallar ('+', '-', '\*', '/') bajarilsin.
8. Shaxsning familiyasi va ismi, tug'ilgan yili, jinsi, yashash manzili va telefon nomeri bo'yicha ma'lumotni xotirada saqlash, qiymat o'qish va chop qilish amalini bajaruvchi **SHAXS** sinfi yaratilsin. Uning vorisi qilib talabalar guruhini tavsivlovchi **TALABA** sinfi yaratilsin. Unda qo'shimcha ravishda talabaning o'qiydigan guruh nomi, kursi haqida ma'lumot bo'lishi kerak. **TALABA** sinfi obektlari chiziqli ro'yxatini yaratgan holda, qandaydir alomat bo'yicha talabani izlash (familiya, tug'ilgan yili yoki telefon nomeri bo'yicha), ro'yxatga yangi yozuvni qo'shish, o'chirish va tartiblash amallari bajarilsin.
9. Natural sonlarning ikkita to'plamini yaratish, chop qilish amallarini o'z ichiga olgan **TUPLAM\_AB** sinfi yaratilsin. To'plam ustida asosiy amallarni – to'plamga yangi element qo'shish va o'chirish, tuplamlar keshishmasini, birlashmasini, hamda ayirmasini bajaradigan funksiya-a'zolari bo'lgan **TUPLAM\_AMALLARI** sinfi **TUPLAM** sinfi vorisi qilib aniqlansin.
10. Berilgan satrni oqimdan o'qish, saqlash, chop qilish amallarini bajaradigan **MATN** sinfi yaratilsin. Uning vorisi sifatida faqat lotin harfida yozilgan matnni shifrlaydigan va qayta tiklaydigan **SHIFRLASH** sinfi aniqlansin. Shifrlash uchun lotin harflar alfaviti olinadi. Jarayon matndagi har bir harf

bo'yicha chapdan o'ng tomonga ketma-ket ravishda amalga oshiriladi. Har qadamda alfavitni ko'rsatilgan songa siklik chapga suriladi va matndagi ayni harfni uning hosil bo'lgan alfavitdagi o'rnidagi (indexidagi) harf bilan almashtiriladi. Har bir qadam uchun alfavitni surish soni beriladi. Masalan, 5,3,2,4 sonlari berilgan bo'lsin. Birinchi qadamda, y'ani matnning birinchi harfini kodlashda alfavit 5 marta chapga siklik suriladi va hosil bo'lgan alfavitdagi qaralayotgan harf o'rnida paydo bo'lgan harf bilan birinchi xarf almashtiriladi. Ikkinchi qadamda alfavit yana 3 marta chapga suriladi va hokazo. To'rtinchi qadamdan keyin, ya'ni alfavit 4 marta chapga surilgandan keyin, surilishlar ro'xati takrorlanadi.

11. Stekni amalga oshiruvchi **STEK** sinfi aniqlansin. Unda stekni tozalash, unga qiymat joylashtirish, o'chirish amallari bajarilsin. Ushbu sinfnin vorisi bo'lgan **LABIRINT** sinfidan labirintdan chiqish masalasini yechishda foydalanilsin. Labirint kvadratlardan tashkil topgan matritsa ko'rinishida beriladi. Har bir kvadrat ochiq yoki yopiq bo'ladi. Yopiq kvadratga kirish mumkin emas. Agar kvadrat ochiq bo'lsa uning yon tomonidan kirish mumkin (burchagidan kirish mumkin emas). Har bir kvadrat uning matritsadagi koordinatalari bilan beriladi. Labirintdan chiqish amalga oshirilganda topilgan yo'l chop qilinadi (kvadratlar koordinatalari juftliklarining ketma-ketligi ko'rinishida).
12. **TAXTA** sinfi shaxmat taxtasini tavsivlaydi. Shaxmat katagi ikkita belgidan tashkil topgan katak ko'rinishida berilgan: lotin harfi (**a** dan **h** gacha) va raqam (1 dan 8 gacha), masalan **a2** yoki **g5**. Ularni farzin joylashgan shaxmat taxtasidagi katak koordinatalari sifatida qarab, farzin "*uradigan*" kataklarni 'X', boshqa kataklarni 'O' bilan belgilab, shaxmat taxtasining ko'rinishi chop qilish imkonini beruvchi voris **FARZIN** sinfi aniqlansin.
13. Ko'rsatkich asosida yaratilgan butun sonlardan iborat navbatni tavsivlovchi **NAVBAT** sinfida navbat bilan ishlash, ya'ni elementlar oxiriga qo'shish, boshidan o'chirish ("*birinchi kelgan–birinchi ketadi*") bilan bog'liq quyidagi funktsiya aniqlanishi zarur bo'ladi:
  - **Tozalash()** - bo'sh navbatni yaratuvchi (navbatni tozalovchi);
  - **BushNavbat()** - navbatni bo'shligini tekshiruvchi;
  - **Navbatga()** - navbat oxiriga yangi element qo'shuvchi;
  - **Navbatdan()** - navbatdagi birinchi elementni qaytaruvchi va uni navbatdan o'chiruvchi.

**NAVBAT** sinfinin vorisi sifatida sonlarning umumiy navbatidan sonlarni musbat sonlar navbatiga va musbat bo'lmagan sonlar navbatiga ajratuvchi **MUSBAT\_MANFIY\_NAVBATLAR** sinfi aniqlansin.
14. Tasodifiy son hosil qiluvchisini shakllarning yuzasini va hajmini hisoblashda qo'llash mumkin. Shunday usullardan birini Monte-Karlo usuli deyiladi (**MONTE\_KARLO** sinfi) va uning mohiyati quyidagicha: faraz qilaylik, **M** shakl birlik kvadrat ichida to'laligicha yotibdi. Tasodifiy son hosil qiluvchisi yordamida birlik kvadrat ichida **n** son tanlandi, agar **v(n)** orqali bu sonlarning **M** shakl ichiga tushganlari miqdorini belgilaylik. U

holda  $M$  shakl yuzasi taqriban  $\frac{v(n)}{n}$  qiymatiga teng bo'ladi va  $n$  qanchalik ko'p bo'lsa yuzaning haqiqiy qiymatiga yaqinlashiladi. Tassodifiy tanlangan nuqta sifatida  $(r_1, r_2), (r_3, r_4), \dots$  koordinatalari bilan berilgan nuqtalarni olish mumkin, bu yerda  $r_1, r_2, \dots$  tassodifiy son hosil qiluvchisi tomonidan olingan sonlar. Xuddi shunday, uch o'lchamli fazodagi nuqtalarni  $(r_1, r_2, r_3)$  koordinatalari bilan tanlash orqali birlik kub ichidagi shakl hajmini hisoblash mumkin.

MONTE\_KARLO sinfining vorislari sifatida analitik ko'rinishi bilan berilgan tekislik figurasi yuzasini hisoblovchi MONTE\_KARLO\_2D, hamda uch o'lchamli fazoda figura hajmini hisoblovchi MONTE\_KARLO\_3D sinfi aniqlansin.

15. Bitta qurilmadan ikkinchisiga kanal orqali 0 va 1 raqamlaridan iborat xabar jo'natayotganda halal beruvchi shovqinlar ta'sirida xabar xato qabul qilinishi mumkin (0 o'rniga 1 yoki 1 o'rniga 0). Bunday xatolikni bartaraf qilish yo'llaridan biri – har bir uzatiladigan raqamlarni uch marta takrorlashdir. Masalan, 1,0,1 xabari 1,1,1,0,0,0,1,1,1 ko'rinishida uzatiladi. Qabul qilishda esa har bir uchta raqamlar guruhi unda eng ko'p uchragan raqam bilan almashtiriladi orqali xabar tiklanadi.

Yuqorida keltirilgan usul bilan berilgan matnni (satrni) “*junatadigan*“ va “*qabul*“ qiladigan amallarni bajaruvchi HABAR tayanch sinfi yaratilsin. Uning vorisi qilib berilgan matnni yuqorida keltirilgan usulda shifrlash orqali jo'natadigan va qabul qiladigan XABARNI\_SHIFRLASH voris sinfi yaratilsin. Bu yerda matn belgilarining ASCII kodlari asosida jo'natiladi, qabul qilinadi va tiklanadi.

16. Haqiqiy son kompyuter xotirasida ko'rinishidagi formatda saqlanadi.

S	P	M
---	---	---

Bu yerda S-son ishorasini aniqlaydi. Agar son musbat bo'lsa  $S=0$ , aks holda  $S=1$  bo'ladi. P-son tartibi (Q-sanoq sistemasi asosining darajasi). M-mantissa ( $0 < M < 1$ ). Har qanday ixtiyoriy son  $(-1)^S M \cdot Q^P$  ko'rinishiga keltirilib saqlanadi.

Oldindan berilgan format o'lchamlari M, P qiymatlariga ko'ra berilgan N sonining ichki formatini tavsiflovchi FLOAT\_FORMAT sinfi aniqlansin. Sonning ichki formatiga mos baytlardagi sonlarni mos ravishda 2, 10, va 16 sanoq sistemasida chop qiluvchi FFORMAT2, FFORMAT10 va FFORMAT16 hosilaviy sinflar yartilsin.

17. Tayanch UCHBURCHAK sinfi aniqlangan, bunda uchburchak uchlarining koordinatalari bilan aniqlangan. Tayanch sinfda Turi()- uchburchak turini aniqlovchi, Yuza() - uchburchak yuzasini hisoblovchi, Perimetr() - uchburchak perimetrni hisoblovchi funksiya-a'zolari bo'lsin.

Quyidagilar amalga oshirilsin:

- a) T\_UCHBURCHAK nomli to'g'ri uchburchak xususiyatlarini o'z ichiga olgan voris sinf qurilsin va uning funksiya-a'zolari aniqlansin;

- b) TT\_UCHBURCHAK nomli teng tomonli uchburchak xususiyatlarini o‘z ichiga olgan voris sinf qurilsin va uning funksiya-a‘zori aniqlansin;
- d) TY\_UCHBURCHAK nomli teng yonli uchburchak xususiyatlarini o‘z ichiga olgan voris sinf qurilsin va uning funksiya-a‘zori aniqlansin.
18. Tayanch TO\_RTBURCHAK sinfida to‘rtburchak uchlari  $A(x_1, y_1)$ ,  $B(x_2, y_2)$ ,  $C(x_3, y_3)$  va  $D(x_4, y_4)$  koordinatalari bilan aniqlangan. Tayanch sinfda Mavjud() – to‘rtburchak mavjudligini aniqlovchi, Yuza() – to‘rtburchak yuzani hisoblovchi, Perimetr() – to‘rtburchak perimetrini hisoblovchi funksiya-a‘zolari aniqlansin. Quyidagi masallar yechilsin:
- a) ROMB nomli romb xususiyatlarini o‘z ichiga olgan voris sinf qurilsin va uning funksiya-a‘zolari aniqlansin;
- b) PARALLELOGRAM nomli parallelogram xususiyatlarini o‘z ichiga olgan voris sinf qurilsin va uning funksiya-a‘zolari aniqlansin;
- d) KVADRAT nomli kvadrat xususiyatlarini o‘z ichiga olgan voris sinf qurilsin va uning funksiya-a‘zolari aniqlansin.
19. Matnni o‘qish, saqlash va chop qilish amallairini o‘z ichiga olgan MATN sinfi aniqlansin. Uning vorisi qilib berilgan matnni formula ekanligini aniqlaydigan FORMULA sinfi yaratilsin. Matnni “formula” ekanligini quyidagi grammatik qoidalar aniqlaydi:
- <foimula>::=<term>|(<formula><ishora><formula>)
- <ishora>::= +|-|\*
- <term>::= <nom>|<butun>
- <nom>::=<harf>|<nom><harJ>|<nom><raqam>
- <butun>::=<raqam>|<butun><raqam>
- <harf>::=a|b|c|d|e|f
- <raqam>::=0|1|2|3|4|5|6|7|8|9
- Nuqta bilan tugaydigan matnning “formula” yoki yo‘qligi aniqlansin.
20. Berilgan sonlar ustida arifmetik amallarni (+, -, \*, /) bajaruvchi KALK\_ARIFMETIKA sinfi aniqlansin. Uning vorisi qilib qo‘shimcha ravishda ln(),  $x^y$ , sin(), cos(), tg(), ctg() va sqrt() funksiyalarini o‘z ichiga olgan KALK\_MUHANDIS sinfi aniqlansin.

## 25. Operatorlarni qayta yuklash

### *Namunaviy masala*

Koeffisientlari va ozod hadi bilan berilgan  $AX=B$  chiziqli tenglamalar sistemasi Gauss usulida yechilsin.

### *Yechish usuli*

Chiziqli tenglamalar sistemasini echishning Gauss usuli haqidagi ma’lumotni hisoblash matematikasi bo‘yicha adabiyotda tanisihish mumkin. Qo‘yilgan masalani echishda Gauss usulining ustun bo‘yicha maksimal elementni tanlash bilan bog‘liq varianti ishlatiladi.

Qo‘yilgan masalani yechish uchun quyidagi sinflar yaratildi:

**Vektor** sinfi - bu berilgan  $n$  o'lchamdagi chiziqli tenglamalar sistemasi  $X$  va  $B$  vektorlarini saqlash, yaratish (oddiy va nusxalash konstruktorlari orqali), vektor elementiga murojaat qilish (`[]` amalini qayta yuklash orqali), vektorni chop qilish (`>>` amalini qayta yuklash orqali) va dinamik massivni o'chirish funksiyalarini o'z ichiga olgan sinf;

**Matritsa** sinfi - berilgan  $n \times n$  o'lchamdagi kvadrat matritsani saqlash (konstruktor vositasida) va matritsa o'lchamini qaytaruvchi funksiy-a'zolariga ega sinf;

**Gauss** sinfi - bu **Matritsa** sinfi vorisi bo'lib unga **Vektor** sinfi agregatsiya qilingan, y'ani sinf tarkibiga **Vektor** turidagi ob'ektlar ( $X$  va  $B$  massivlar) kiritilgan. Bunda tashqari sinf tarkibiga konstruktor orqali chiziqli tenglamalar sistemasining o'lchamini, matritsa qiymatlarini ( $A$  matritsa) va ozod hadlar ( $B$  vektorni) kiritish, hamda Gauss usulini amalga oshiruvchi **Metod()** funksiya-a'zo, tenglamalar sistemasi echimini chop qiluvchi (`>>` amalini qayta yuklash orqali) va olingan echimni tenglamalar sistemasini qanoatlantirishini tekshirish uchun matritsaning boshlangich holatini tiklash uchun (`<<` amalini qayta yuklash orqali) operator-funksiyalar kiradi.

**Matritsa** va **Vektor** sinfida qiymatlarni o'qish va chop qilishda funksiyalar parametri sifatida uzatiladigan vizual komponentalardan foydalanilgan.

Sinflar e'lonlari `<gauss.h>` sarlavha fayliga yoziladi. `Gauss.cpp` modulida **Gauss** turidagi ko'rsatkich bo'yicha obyekt yaratiladi va unga `TStringGrid` obyektlaridan qiymatlar uzatiladi.

Ilova bilan ishlash qadamlari (quyida keltirilgan shaklga qarang):

1. "**Matritsa o'lchami  $N$ :**" nishoni bilan belgilangan bir qatorli tahrir maydoniga sistema o'lchami -  $n$  soni kiritiladi va "`cin>>N`" tugmasi bosiladi.

2. Kiritilgan  $n$  soniga mos ravishda `TStringGrid` turidagi "**A matritsa**" hamda "**B vektor**" o'lchamlari shakllanadi va ularning kataklariga qiymatlar kiritiladi.

3. Sistema koeffisientlari kiritilgandan keyin "**Gauss**" tugmasi bosiladi. Agar sistema echimga ega bo'lsa u "**X vektor**" bilan belgilangan kataklarga chop etiladi, aks holda echim yo'qligi haqida xabar beriladi.

4. Olingan  $X$  yechimni tenglamalar sistemasini qanoatlantirishini tekshirish uchun "**Tekshirish**" tugmasi bosiladi va ozod hadlar -  $B$  vektor qiymatlari "**B1 vektor**" kataklariga chop etiladi.

Shuni qayd etish kerakki, programmada qiymatlar kiritishlarini nazorat qilish va tenglamalar sistemasi qiymatlarini qaytadan kiritish holatlari inobatga olinmagan.

### ***Programma matni***

#### ***Unit\_Gauss.h fayli:***

...

```
enum TStringdoubleFormat {sffGeneral, sffExponent, sffFixed,
sffNumber, sffCurrency };
class Matritsa
```



```

{
    int na;
protected:
    double **A;
public:
    Matritsa(int _na, TStringGrid * SG);
    ~Matritsa();
    int n_matritsa(){return na;}
};
Matritsa::Matritsa(int _na, TStringGrid * SG)
{
    na=_na;
    A=new double*[na];
    for(int i=0;i<na;i++)
        A[i]=new double[na];
    for(int i=0;i<na;i++)
        for(int j=0;j<na;j++)
            A[i][j]=StrToFloat(SG->Cells[j][i]);
}
Matritsa::~Matritsa()
{
    for(int i=0;i<na;i++)
        delete [] A[i];
    delete A;
}
class Vektor
{
    int nv;
    double *v;
public:
    Vektor(int,TStringGrid*);
    Vektor(Vektor*);
    ~Vektor(){delete [] v;}
    void operator>>(TStringGrid*);
    int n_vektor(){return nv;}
    double & operator[](int i){return v[i];}
};
Vektor::Vektor(int _nv,TStringGrid * SG)
{
    nv=_nv;
    v=new double[nv];
    for(int i=0;i<nv;i++) v[i]=StrToFloat(SG->Cells[0][i]);
}
Vektor::Vektor(Vektor * V)

```

```

{
if(V)
{
    nv=V->n_vektor();
    v=new double[nv];
    for(int i=0;i<nv;i++)v[i]=V->operator [](i);
}
}
void Vektor::operator>>(TStringGrid *SG_v)
{
for(int i=0;i<nv;i++)
    SG_v->Cells[0][i]=FloatToStrF(v[i],sffFixed,6,2);
}
class Gauss:public Matritsa
{
    Vektor *B,*X;
public:
    Gauss(int n,TStringGrid* SG_A,TStringGrid* SG_B): Matritsa(n,SG_A)
    { B=new Vektor(n,SG_B);}
    ~Gauss(){ delete [] B; delete [] X;}
    bool Metod();
    Vektor * AX_B();
    void operator >>(TStringGrid*);
    void operator <<(TStringGrid*);
    Vektor * Vektor_B(){return B;}
};
bool Gauss::Metod()
{
    double r;
    int k;
    int N=n_matritsa();
    X=new Vektor(B);
    for(int i=0;i<N;i++)
    {
        k=i;
        r=fabs(A[i][i]);
        for(int j=i+1;j<N;j++)
            if(fabs(A[j][i])>r)
            {
                k=j;
                r=fabs(A[j][i]);
            }
        if(r==0.0) return 0;
        if(k!=i)

```

```

{
r=X->operator [](k);
X->operator [](k)=X->operator [](i);
X->operator [](i)=r;
for(int j=i;j<N;j++)
{
r=A[k][j];
A[k][j]=A[i][j];
A[i][j]=r;
}
}
r=A[i][i];
(*X)[i]=(*X)[i]/r;
for(int j=i;j<N;j++)
A[i][j]=A[i][j]/r;
for(int k=i+1;k<N;k++)
{
r=A[k][i];
(*X)[k]=(*X)[k]-r*(*X)[i];
for(int j=i;j<N;j++)
A[k][j]=A[k][j]-r*A[i][j];
}
}
for(int i=N-2;i>=0;i--)
for(int j=i+1;j<N;j++)
(*X)[j]=(*X)[j]-A[i][j]*(*X)[i];
return 1;
}
Vektor* Gauss::AX_B()
{
for(int i=0;i<n_matritsa();i++)
{
(*B)[i]=0;
for(int j=0;j<n_matritsa();j++)
(*B)[i]+=A[i][j]*(*X)[j];
}
return B;
}
void Gauss::operator >>(TStringGrid * Sg_X)
{
for(int i=0;i<X->n_vektor();i++)
Sg_X->Cells[0][i]=FloatToStrF((*X)[i],sffFixed,6,2);
}
void Gauss::operator <<(TStringGrid * SG_A)

```

```

{
for(int i=0;i<n_matritsa();i++)
for(int j=0;j<n_matritsa();j++)
A[i][j]=StrToFloat(SG_A->Cells[j][i]);
}
Gauss *gauss;

```

*Unit\_Gauss.cpp fayli:*

```

...
int n;
void __fastcall TForm1::Btn_UlchamClick(TObject *Sender)
{
n=StrToInt(Edit_n->Text);
StringGrid_A->RowCount=n; StringGrid_A->ColCount=n;
StringGrid_B->RowCount=n; StringGrid_B->ColCount=1;
StringGrid_X->RowCount=n; StringGrid_X->ColCount=1;
StringGrid_BT->RowCount=n; StringGrid_BT->ColCount=1;
}
void __fastcall TForm1::Btn_GaussClick(TObject *Sender)
{
gauss=new Gauss(n,StringGrid_A,StringGrid_B);
if(gauss->Metod())(*gauss)>>StringGrid_X;
else
ShowMessage("Tenglamalar sistemasi echimga ega emas.");
}
void __fastcall TForm1::Btn_TestClick(TObject *Sender)
{
*gauss<<StringGrid_A;
*gauss->AX_B()>>StringGrid_BTest;
}

```

Ilova shaklining (formaning) ko‘rinishi quyidagi rasmda keltirilgan

**AX=B Chiziqli tenglamalar sistemasini Gauss usulida yechish**

Matritsa o'lchami N:

A matritsasi			B vektor	X vektor	B1 vektor
4	0,24	-0,08	8	1,91	8,00
0,09	3	-0,15	9	3,19	9,00
0,04	-0,08	4	20	5,04	20,00

### ***Amaliy topshiriqlar***

1. Kompleks sonlar ustida arifmetik amallar operator-funksiya qilib aniqlangan **KOMPLEKS** sinfi yaratilib, uning obyektlari ustida mos amallar bajarilsin.
2. Berilgan  $n$  o'lchamli vektor ustida vektorlarni qo'shish, ayirish, skalyar ko'paytirish, hamda vektorni songa ko'pytirish amallari qayta yuklangan **VEKTOR** sinfi aniqlansin. Sinf obyektlari ustida ko'rsatilgan amallar bajarilsin.
3. Berilgan natural  $n$  va  $m$  o'lchamdagi haqiqiy elementli matritsa uchun xotiradan joy ajratish, qiymatlarini o'qish va chop qilish amallarini bajaradigan **MATRITSA** tayanch sinfi yaratilsin. Berilgan  $A$  va  $B$  matritsalar ustida  $A+B$ ,  $A-B$ ,  $A*B$ , amallarini bajaradigan **ARIFM\_MATRITSA** sinfi **MATRITSA** sinfidan voris sifatida yaratilsin va unda ko'rsatilgan amallar qayta yuklansin.
4. Vektor yordamida to'plamni hosil qilish amalinini bajaruvchi **TUPLAM** sinfi yaratilsin. To'plam ustida asosiy amallarni - to'plamga yangi element qo'shish va o'chirish, to'plamlar keshishmasi, birlashmasi, hamda ayirmasi amallari qayta yuklansin. Sinf obyektlari ustida ko'rsatilgan amallar bajarilsin.
5. Vaqtning sekunt, minut, soat qiymatlari ustida bajariladigan qo'shish, ayirish va taqqoslash amallari qayta yaklaydigan **VAQT** sinfi aniqlansin. Sinf obyektlari ustida ko'rsatilgan amallar bajarilsin.
6. Sananing kun, oy, yil qiymatlari ustida bajariladigan qo'shish, ayirish va taqqoslash amallari qayta yaklaydigan **SANA** sinfi aniqlansin. Sinf obyektlari ustida ko'rsatilgan amallar bajarilsin.
7. Rasional sonlar ustida, yani surat va mahraj juftligi bilan berilgan sonlar ustida qo'shish, ayirish, kopaytirish taqqoslash amallarini qayta yaklaydigan **RATSIONAL** sinfi aniqlansin. Sinf obyektlari ustida ko'rsatilgan amallar bajarilsin.

8. Dekart koordinatasida, tekislikda berilgan nuqta koordinatasini Qutb koordinatasiga va aksincha, Qutbdan koordinatasidan Dekart koordinatasiga otkazuvchi amallarni o'z ichiga olgan QUTB va DEKART sinflari aniqlansin. Sinflar obyektlari ustida ko'rsatilgan amallar bajarilsin.
9. AKSLANTIRISH\_01 sinfi aniqlansin. Unda haqiqiy sonlar massivini  $[0,1]$  segmentga akslantirish operator – funksiya ko'rinishida aniqlansin. Sinf obyektlari ustida ko'rsatilgan amallar bajarilsin.
10. SATR sinfi aniqlansin va unda nol terminalli satrlar ustida satrga satr qo'shish, satrdagi bir satr ostini ikkinchi satr bilan almashtirish amallari operator-funksiya ko'rinishida aniqlansin. Sinf obyektlari ustida ko'rsatilgan amallar bajarilsin.
11. STEK sinfi aniqlansin. Unda stekga element joylash va o'qish, stek bo'shligini tekshirish amallari operator-funksiya sifatida aniqlansin. Sinf obyektlari ustida ko'rsatilgan amallar bajarilsin.
12. Haqiqiy sonlarning  $[a,b]$  va  $[c,d]$  yopiq intervallari ustida quyidagi amallar aniqlangan:  
 $[a,b]+[c,d]=[a+c,b+d];$   
 $[a,b]-[c,d]=[a-c,b-d];$   
 $[a,b]*[c,d]=[\min(a*c,a*d,b*c,b*d), \max(a*c,a*d,b*c,b*d)];$   
 $[a,b]/[c,d]=[a,b]*[1/c,1/d]$   $c>0$  yoki  $d<0$  shartida;  
 $[a,b]==[c,d]$ , agar  $a=c$  va  $b=d$ ;  
 $[a,b]<=[c,d]$  agar  $a\leq c$  va  $b\leq d$ .  
 INTERVAL sinfi aniqlanib, unda yuqoridagi amallari qayta yuklansin. Sinf obyektlari ustida ko'rsatilgan amallar bajarilsin.
13. Chekli  $n$  o'lchamli mantiqda qiymatlar 0 dan  $n-1$  gacha sonlar bo'lib, undagi amallar quyidagicha aniqlanadi:  
 $a*b=\min(a,b);$   
 $a+b=\max(a,b);$   
 $a-b=a+n-1-b.$   
 N\_MANTIQ sinfi aniqlansin. Unda yuqoridagi amallar qayta yuklansin. Sinf obyektlari ustida ko'rsatilgan amallar bajarilsin.
14. Ishoralar mantig'ida amallar quyidagicha aniqlanadi:  
 $a*b=\text{sign}(a)*\text{sign}(b);$   
 $a+b=\text{sign}(a)+\text{sign}(b);$   
 $a-b=\text{sign}(a)-\text{sign}(b).$   
 ISHORA\_MANTIQ sinfi aniqlanib? unda yuqoridagi amallar qayta yuklansin. Sinf obyektlari ustida ko'rsatilgan amallar bajarilsin.
15. Berilgan  $n$ -darajali  $P(x)$  ko'phadini tavsiflovchi KUPHAD sinfi yaratilsin. Unda ikkita ko'phadni qo'shish, ko'pytirish, ayirish amallari, hamda berilgan  $a$  va  $n$  uchun  $P(a)$  qiymatini  $m$  soni bilan taqqoslash amallari opertor - funksiya ko'rinishida aniqlansin.
16. Katta sonlar ustida arifmetik amallarni bajarish. O'nlik sanoq sistemasidagi ikkita  $a$  va  $b$  butun sonlar satr ko'rinishida berilgan.  $a+b$ ,  $a-$

$b$ ,  $a*b$  va  $a/b$  amallar qayta yuklanuvchi operator ko‘rinishida aniqlangan UZUN\_SON sinfi aniqlansin.

17. Uzunligi oldindan noma'lum bo'lgan binar  $a$  va  $b$  sonlar ustida arifmetik amallar bajarilsin. Sonlar satr ko‘rinishida berilgan.  $a+b$ ,  $a-b$ ,  $a>>n$  ( $a$  razryadlarini o‘ngga  $n$  pozitsyaga surish),  $a<<n$  ( $a$  razryadlarini o‘ngga  $n$  pozitsyaga surish) va  $a^b$  (istisnoli yoki) amallari qayta yuklanuvchi operator ko‘rinishida aniqlangan BINAR\_SON sinfi aniqlansin.
18.  $0 \leq a \leq 11, 0 \leq b \leq 11$  butun sonlar ustida qoshish amali gadvalda berilgan huddi shunday ayirish amalini ham aniqlash mumkin, bunday amallarga modulyar arifmetika deyiladi. Huddi shu amallar aniqlangan SOAT\_ARIFMETIKASI sinfi aniqlansin.

0	1	2	3	4	5	6	7	8	9	10	11
1	2	3	4	5	6	7	8	9	10	11	0
2	3	4	5	6	7	8	9	10	11	0	1
3	4	5	6	7	8	9	10	11	0	1	2
4	5	6	7	8	9	10	11	0	1	2	3
5	6	7	8	9	10	11	0	1	2	3	4
6	7	8	9	10	11	0	1	2	3	4	5
7	8	9	10	11	0	1	2	3	4	5	6
8	9	10	11	0	1	2	3	4	5	6	7
9	10	11	0	1	2	3	4	5	6	7	8
10	11	0	1	2	3	4	5	6	7	8	9
11	0	1	2	3	4	5	6	7	8	9	10

## 26. Polimorfizm

Ushbu bo‘limda qo‘yilgan masala mazmunidan kelib chiqqan holda uning umumiy xususiytlarini o‘z ichiga olgan tayanch sinf yaratiladi, undan voris sifatida hosil bo‘lgan sinf orqali bevosita qo‘yilgan masala yechiladi.

### *Namunaviy masala*

Shaxsning o‘rta ma’lumot yoki oliy ma’lumot darajasiga mos ravishda anketa ma’lumoti to‘ldirilsin.

### *Yechish usuli*

Shaxsning o‘rta yoki oliy ma’lumotga ega ekanligiga mos ravishda uning anketa punktlari turlicha bo‘ladi. Masalan, o‘rta ma’lumotli shaxsning anketasiga quyidagi maydonlar kirishi mumkin: ismi, familiyasi, maktab (litsey/kollej) nomeri va maktabni (litsey/kollejni) tugatgan yili. O‘z navbatida oliy ma’lumotli shaxs uchun esa bu maydonlar quyidagicha bo‘lishi mumkin: ismi, familiyasi, OTM nomi, fakultet nomi, OTMni tugatgan yili va mutaxassisligi.

Qo‘yilgan masala ikkita o‘zaro vorislik munosabatida bo‘lgan sinflarni yaratish bilan yechiladi:

– tayanch UMUMIY\_TA\_LIM sinfida oʻrta maʼlumotga ega shaxs uchun anketani shakllantirish, kiritilgan maʼlumotni qabul qilish va chop qilish amallari bajariladi. Maʼlumotni qabul qilish va chop qilish formaning TStringGrid va TMemo turidagi komponentalari vositasida amalga oshiriladi (mos funktsiya-aʼzolarining parametrida koʻrsatilgan).

– voris OLIY\_TA\_LIM sinfida oliy maʼlumotli shaxsning anketasi bilan bogʻliq oʻziga xos amallar aniqlanadi.

Anketaning koʻrinishini mos ravishda shakllantirish va chop qilish uchun polimorfizmdan foydalaniladi. Bu vazifani bajaruvchi funktsiyalar virtual aniqlanadi:

```
virtual void Anketa_shakli(TStringGrid*);  
virtual void Chop_etish(TMemo*);
```

Programma ishlashida quyidagi ketma-ketlikka amal qilinadi:

1. Shaxsning oʻrta yoki oliy maʼlumotga ega ekanligi aniqlanadi;
2. Taʼlim turiga mos anketa shakllantiriladi;
3. Anketa maʼlumotlari qabul qilinadi va chop etiladi.

### ***Programma matni***

#### ***Unit\_Anketa.h fayli:***

```
...  
class UMUMIY_TA_LIM  
{  
public:  
    UMUMIY_TA_LIM();  
    ~UMUMIY_TA_LIM();  
    void Anketa_qabul(TStringGrid*);  
    virtual void Anketa_shakli(TStringGrid*);  
    virtual void Chop_etish(TMemo*);  
protected:  
    int n;  
    String * shaxs;  
};  
UMUMIY_TA_LIM::UMUMIY_TA_LIM()  
{ n=4; shaxs=new String[n]; }  
UMUMIY_TA_LIM::~~UMUMIY_TA_LIM()  
{ delete []shaxs; }  
void UMUMIY_TA_LIM::Anketa_shakli(TStringGrid * SG)  
{  
    SG->RowCount=n;  
    SG->Cells[0][0]="Ismi:";  
    SG->Cells[0][1]="Familiyasi:";  
    SG->Cells[0][2]="Maktab(L/K) nomeri:";  
    SG->Cells[0][3]="Maktab(L/K) tugatgan yili:";
```



```

}
void UMUMIY_TA_LIM::Anketa_qabul(TStringGrid * SG)
{
    for(int i=0; i<n;i++) shaxs[i]=SG->Cells[1][i];
}
void UMUMIY_TA_LIM::Chop_etish(TMemo *memo)
{
    memo->Lines->Clear();
    memo->Lines->Add(" O'RTA TA'LIMGA EGA SHAXS: ");
    memo->Lines->Add("*****");
    memo->Lines->Add("Ismi : "+shaxs[0]);
    memo->Lines->Add("Familiyasi: "+shaxs[1]);
    memo->Lines->Add("Maktab(L/K) nomeri: "+shaxs[2]);
    memo->Lines->Add("Maktab(L/K) tugatgan yili: "+shaxs[3]);
}
class OLIY_TA_LIM :public UMUMIY_TA_LIM
{
public:
    OLIY_TA_LIM();
    ~OLiy_TA_LIM();
    void Anketa_shakli(TStringGrid*);
    void Chop_etish(TMemo*);
};
OLiy_TA_LIM::OLiy_TA_LIM()
{
    n=6; shaxs=new String[n];
}
OLiy_TA_LIM::~~OLiy_TA_LIM()
{
    delete []shaxs;
}
void OLIY_TA_LIM::Anketa_shakli(TStringGrid * SG)
{
    SG->RowCount=n;
    SG->Cells[0][0]="Ismi:";
    SG->Cells[0][1]="Familiyasi:";
    SG->Cells[0][2]="OTM nomi:";
    SG->Cells[0][3]="Fakultet nomi:";
    SG->Cells[0][4]="OTM tugatgan yili:";
    SG->Cells[0][5]="Mutaxassisligi:";
}
void OLIY_TA_LIM::Chop_etish(TMemo *memo)
{
    memo->Lines->Clear();

```

```

memo->Lines->Add(" OLIY TA'LIMGA EGA SHAXS: ");
memo->Lines->Add("*****");
memo->Lines->Add("Ismi : "+shaxs[0]);
memo->Lines->Add("Familiyasi: "+shaxs[1]);
memo->Lines->Add("OTM nomi: "+shaxs[2]);
memo->Lines->Add("Fakultet nomi: "+shaxs[3]);
memo->Lines->Add("OTM tugatgan yili: "+shaxs[4]);
memo->Lines->Add("Mutaxassisligi: "+shaxs[5]);
}
UMUMIY_TA_LIM * Shaxs;

```

*Unit\_Anketa.cpp fayli:*

```

...
void __fastcall TForm1::Btn_Ta_limClick(TObject *Sender)
{
    if(RadioGroup_talim_turi->ItemIndex==0)
        Shaxs=new UMUMIY_TA_LIM;
    else Shaxs=new OLIY_TA_LIM;
    Shaxs->Anketa_shakli(StringGrid_Anketa);
}
void __fastcall TForm1::Btn_Anketa_QabulClick(TObject *Sender)
{
    Shaxs->Anketa_qabul(StringGrid_Anketa);
}
void __fastcall TForm1::Btn_Anketa_ChopClick(TObject *Sender)
{
    Shaxs->Chop_etish(Memo_Anketa);
}

```

Programmaning ish holatlari quyidagi rasmlardada keltirilgan.

1. O'rta ma'lumotli shaxs uchun      2. Oliy ma'lumotli shaxs uchun

The image displays two side-by-side screenshots of a Windows application titled "SHAXS ANKETASI".

**Left Screenshot (O'rta maxsus ta'lim):**

- Shaxs ta'limi:** Radio buttons for "O'rta maxsus ta'lim" (selected) and "Oliy ta'lim".
- Ta'lim turini tanlash:** A button to the right of the radio buttons.
- Anketa ma'lumotlarini kiritish:** A table with the following data:
 

Ismi:	Akmaljon
Familiyasi:	Madraximov
Maktab(L/K) nomeri:	101
Maktab(L/K) tugatgan yili:	2002
- Malumotlarni qabul:** A button to the right of the table.
- Anketa malumotlari:** A text area showing the entered data: "O'RTA TA'LIMGA EGA SHAXS: Ismi : Akmaljon, Familiyasi: Madraximov, Maktab(L/K) nomeri: 101, Maktab(L/K) tugatgan yili: 2002".
- Ma'lumotni chop\_Qilish:** A button at the bottom right.

**Right Screenshot (Oliy ta'lim):**

- Shaxs ta'limi:** Radio buttons for "O'rta maxsus ta'lim" and "Oliy ta'lim" (selected).
- Ta'lim turini tanlash:** A button to the right of the radio buttons.
- Anketa ma'lumotlarini kiritish:** A table with the following data:
 

Ismi:	Shavkat
Familiyasi:	Madraximov
OTM nomi:	ToshDU
Fakultet nomi:	TMM
OTM tugatgan yili:	1983
Mutaxassisligi:	Amaliy matematika
- Malumotlarni qabul qilish:** A button to the right of the table.
- Anketa malumotlari:** A text area showing the entered data: "OLIY TA'LIMGA EGA SHAXS: Ismi : Shavkat, Familiyasi: Madraximov, OTM nomi: ToshDU, Fakultet nomi: TMM, OTM tugatgan yili: 1983, Mutaxassisligi: Amaliy matematika".
- Ma'lumotni chop\_Qilish:** A button at the bottom right.

### *Amaliy topshiriqlar*

1. Sonlarning progressiya ketma-ketligi hosil qiluvchi **PROGRESSIYA** abstrakt sinfi aniqlansin. Uning vorisi sifatida arifmetik va geometrik progressiyalar ustida amal bag'aruvchi **ARIFM\_PROGRESS** va **GEOM\_PROGRESS** sinflari aniqlansin. Undagi progressiya hadini topish, berilgan  $n$  sondagi progressiya hadlarini yig'indisini hisoblash funksiya-a'zolari virtual qilib aniqlansin.
2. 10 lik sanoq sistemasida berilgan sonni o'qish, saqlash va chop qilish amallarini o'z ichiga olgan tayanch **SANOQ\_SISTEMA\_10** sinfi va uning vorisi sifatida berilgan sonni 2, 8 va 16 lik sanoq sistemasidagi ko'rinishini chop qiluvchi **Sonni\_chop\_qilqish()** virtual funksiya-a'zosi bo'lgan **SANOQ\_SIS\_2**, **SANOQ\_SIS\_8** va **SANOQ\_SIS\_16** sinflar aniqlansin.
3. Kvadrat tenglama ildizlarini hisoblaydigan tayanch **KVADRAT**, uning vorisi **BIKVADRAT** sinflar yaratilsin va ularda tenglama ildizini topadigan **Kv\_ildiz()** polimorf funksiya aniqlansin.
4. Matritsa uchun **MATRITSA** tayanch sinfi yaratilsin. Uning vorisi sifatida to'g'riburchakli (**TB\_MATRITSA**) va kvadrat (**KB\_MATRITSA**) matritsalar ustida qo'shish, ayirish va ko'paytirish amalarni bajaradigan voris sinflar yaratilsin. Kvadrat matritsa elementlari bosh diagonalga nisbatan simmetrik va u uchburchak ko'rinishda berilgan. Yuqoridagi amallarni bajaruvchi funksiyalar polimorf qilib aniqlansin.
5.  $n$  o'lchovli fazoda koordinatalari bilan berilgan ikki nuqta orasidagi masofani hisoblaydigan **MASOFA** tayanch sinfi aniqlansin. Uning vorisi sifatida Dekart, Chebishev va Hemming fazosida nuqtalar orasidagi masofani hisoblaydigan **DEKART**, **CHEBISHEV** va **HEMMING** sinflari yaratilsin. Nuqtalar orasidagi masofani hisoblovchi **roo()** funksiyasi polimorf qilib aniqlansin. Metrikalar quyidagi ko'rinishga ega:
  - Evklid:  $\rho(x, y) = \sqrt{(x_1 - y_1)^2 + \dots + (x_n - y_n)^2}$  ;
  - Chebishev:  $\rho(x, y) = \max_{1 \leq i \leq n} |x_i - y_i|$  ;
  - Xemming:  $\rho(x, y) = \sum_{i=1}^n |x_i - y_i|$  .
6. Butun sonlarning chiziqli ro'yxatini qayta ishlash uchun **RUYXAT** tayanch sinfi yaratilsin. Uning vorisi sifatida stek va navbat tuzulmalari uchun **STEK** va **NAVBAT** sinflari hosil qilinsin va elementlarni ro'yxatga joylash, undan olish amallarini bajaruvchi funksiyalar polimorf tarzda aniqlansin.
7. Yulduzcha va shina topologiyalarida to'r hosil qilish uchun umumiy xarajat hisoblansin. Buning uchun tayanch **TARMOQ** sinfi yaratilsin. Sinfda qurilmalar soni  $n$ , qurilmalargacha bo'lgan masofalar  $a[n]$ , sim narxi  $q$ , konnektor narxi  $p$  berilgan-a'zolari va ularni qayta ishlash funksiya-a'zolari aniqlansin.  
**TARMOQ** sinfidan hosilaviy **YULDUZ** va **SHINA** sinflar yaratilsin.

YULDUZ sinfida xarajatlar  $S = \sum_{k=1}^n (a_k q + 2p)$  formula bilan, SHINA sinfida esa  $S = \max_{1 \leq k \leq n} \{a_k\} q + np$  formula bilan hisoblanadi.

Berilgan topologiya va o'ldamlar bo'yicha S xarajat hisoblansin.

8. Telefon muloqoti xarajatini hisoblovchi TARIF sinfi tuzilsin. TARIF sinfida kiruvchi va chiquvchi qo'ng'iroqlar daqiqalari soni saqlansin. TARIF sinfidan UNIVERSAL va PROGRESS sinflari voris qilib tuzilsin.

Universal (UNIVERSAL) tarifida xarajat  $S = nA + mB$  formula yordamida hisoblanadi. Bu yerda n - kiruvchi daqiqalar soni, m - chiquvchi daqiqalar soni va  $A=0$ ,  $B=0.03\$$  mos ravishda bir daqiqa uchun to'lov narxi.

Progress (PROGRESS) tarifida xarajat  $S = nA + m_1 B_1 + m_2 B_2 + m_3 B_3$  ko'rinishida hisoblanadi. Bu yerda n - kiruvchi daqiqalar soni, m - chiquvchi daqiqalar soni,  $A=0.01\$$ ,  $B_1=0.02\$$ ,  $B_2=0.01\$$ ,  $B_3=0.005\$$ .

Hisoblash shartlari:

- agar  $m \leq 50$  bo'lsa,  $m_1 = m$ ,  $m_2 = m_3 = 0$ ;
- agar  $50 < m \leq 100$  bo'lsa,  $m_1 = 50$ ,  $m_2 = m - 50$ ,  $m_3 = 0$ ;
- agar  $m > 100$  bo'lsa,  $m_1 = 50$ ,  $m_2 = 50$ ,  $m_3 = m - 100$ .

Berilgan tarif, kirish-chiqish qo'ng'iroqlariga ko'ra oylik xarajatlar polimorf funksiyalar orqali hisoblansin.

9. Jismoniy shaxsdan daromad solig'ini olish masalasini yechish uchun quyidagi belgilashlarni kiritilgan (pul birligida): Min\_IH - minimal ish haqi; DM - daromad miqdori; DS - daromad solig'i.

Daromad solig'ini olish qoidalari:

a) imtiyozga ega bo'lmagan shaxslar uchun DM shkalalari:

- 1)  $[1 * \text{Min\_IH}..5 * \text{Min\_IH}]$  pul miqdori uchun daromad solig'i - 9%;
- 2)  $[5 * \text{Min\_IH} + 1..10 * \text{Min\_IH}]$  pul miqdori uchun daromad solig'i - 16%;
- 3)  $10 * \text{Min\_IH}$  dan katta pul miqdori uchun daromad solig'i - 22%.

b) nogiron shaxslar uchun DM shkalalari:

- 1)  $[1 * \text{Min\_IH}..4 * \text{Min\_IH}]$  pul miqdori uchun daromad solig'i - 0%;
- 2)  $[4 * \text{Min\_IH} + 1..5 * \text{Min\_IH}]$  pul miqdori uchun daromad solig'i - 9%;
- 3)  $[5 * \text{Min\_IH} + 1..10 * \text{Min\_IH}]$  pul miqdori uchun daromad solig'i - 16%;
- 4)  $10 * \text{Min\_IH}$  dan katta pul miqdori uchun daromad solig'i - 22%.

Jismoniy shaxsdan olinadigan daromad solig'ini hisoblaydigan SOLIQ sinfi aniqlansin. Uning vorisi sifatida mos ravishda nogironlardan olinadigan daromad solig'ini hisoblovchi SOLIQ\_NOGIRON sinfi aniqlansin. Ikkita sinfda Daromad\_soliq() funksiyasi polimorf qilib aniqlansin.

10. To'rtburchak yuzasini hisoblash uchun TURTBURCHAK tayanch sinfi va romb (ROMB), kvadrat (KVADRAT), to'g'ri to'rtburchak (TTURTBURCHAK), trapetsiya (TRAPETSIYA) va parallelogram (PARALLELOGRAM) voris sinflari yaratilsin va ularda yuzani

hisoblovchi *yuza()* funksiyasi polimorf ko‘rinishda aniqlansin.

11. Berilgan sonlar ketma-ketligini tartiblash uchun **TARTIBLASH** sinfi aniqlasin. Uning vorisi sifatida **PUFAKCHA**, **ORAGA\_JOYLASH** va **PUFAKCHA\_SURISH** sinflari yaratilsin. Unda mos ravishda 16.6, 16.7 larda keltirilgan algoritmlar va “*pufakcha usuli*” algoritmining takomillashgan varianti amalga oshirilsin: massivning yonma-yon joylashgan  $x_i$  va  $x_{i+1}$  elementlari o‘rni almashgandan keyin tartiblanish buzilmagan holda  $x_{i+1}$  element chap tomonga mumkin qadar ko‘chiriladi. Barcha sinflarda tartiblashni amalga oshiradigan virtual **Tartiblash()** funksiya-a’zo aniqlasin.

12. Satr ko‘rinishida berilgan matematik funksiyaaning aniq integralini taqribiy hisoblash uchun quyidagi sinflar aniqlansin:

- **IFODA** sinfi satr-ifodanini qayta ishlash uchun;
- **INTEGRAL** sinfi **IFODA** sinfining vorisi bo‘lib, satr ko‘rinishida berilgan matematik ifodaning qiymati va integralini hisoblash uchun;
- **INTEGRAL\_TRAPETSIYA** sinfi **INTEGRAL** sinfining vorisi bo‘lib, berilgan oraliq uchun satr ko‘rinishidagi funksiya integralini trapetsiya usulida hisoblash uchun;
- **INTEGRAL\_SIMPSON** sinfi **INTEGRAL** sinfining vorisi bo‘lib, berilgan oraliq uchun satr ko‘rinishidagi funksiya integralini Simpson usulida hisoblash uchun.

Sinflarda integralni hisoblash virtual funksiya-a’zo ko‘rinishida amalga oshirilsin.

13. Berilganlarni “*k ta sinflar o‘rtalari*” algoritmi bo‘yicha guruhlash. Algoritm uchun  $m$  satr va  $n$  ustundan iborat matritsa va  $k$  soni boshlang‘ich berilganlar hisoblanadi. Bu yerda matritsaning har bir satri ( $S_1, S_2, \dots, S_m$ )  $n$  o‘lchovli fazodagi nuqtalar,  $k$  – guruhlar soni.

Algoritmning birinchi qadamida  $k$  ta o‘zaro kesishmaydigan  $C_1, C_2, \dots, C_k$  guruhlar markazlari ixtiyoriy ravishda  $S_1, S_2, \dots, S_m$  nuqtalar orasidan tanlanadi.

Guruh tarkibi quyidagi qoida bilan aniqlanadi:  $S_i \in C_j$  bo‘ladi, agar  $\rho(S_i, C_j) = \min_{1 \leq r \leq k} \rho(S_i, C_r)$  bo‘lsa. Bu yerda  $\rho(S_i, C_j)$  - bu  $S_i$  nuqta va  $C_j$  guruh markazi orasidagi masofa.

Guruhlar tarkibi aniqlangandan keyin har bir  $C_v$  guruh uchun uning markazi topiladi. Guruh markazi unga kiruvchi nuqtalar o‘rta arifmetigi bilan aniqlanadi:

$$C_v = \frac{\sum_{S_i \in G_v} S_i}{|G_v|}, v = \overline{1, k}.$$

Keyingi qadamlarda guruhlarning tarkibi va markazi qayta aniqlanadi. Bu jarayonlar iterativ ravishda amalga oshiriladi. Guruhlar tarkibi va markazining o‘zgarmay qolishi hisoblash jarayonini to‘xtatish alomati

bo‘ladi.

$n$  o‘lchamli fazoda berilgan  $x = (x_1, x_2, \dots, x_n)$  va  $y = (y_1, y_2, \dots, y_n)$  nuqtalar orasidagi masofani hisoblashda Dekart, Chebishev va Xemming metrikalaridan foydalanilsin (26.5 masalaga qarang).

Masalani yechish uchun tayanch GURUH sinfi va uning vorislari sifatida GURUH\_EVKLID, GURUH\_XEMMING, GURUH\_CHEBISHEV sinflari aniqlansin. Barcha sinflarda masofa hisoblaydigan virtual RooXY() funksiya-a‘zo bo‘lsin.

14. Tekislikda berilgan sohani diskretlash deb sohanini elementar bo‘laklarga ajratish tushiniladi.

Soha  $(x_1, y_1), (x_2, y_2), (x_3, y_3)$  va  $(x_4, y_4)$  tugun nuqtalari bilan to‘rtburchak k‘rinishida beriladi. Sohani quyidagi usullarda diskretlash mumkin:

1) kvadrat elementlarga bo‘lish usuli. Soha tomonlari o‘zaro teng ikki bo‘lakka bo‘linadi;

2) kub elementlarga bo‘lish usuli. Soha tomonlari o‘zaro teng uchta bo‘laklarga bo‘linadi.

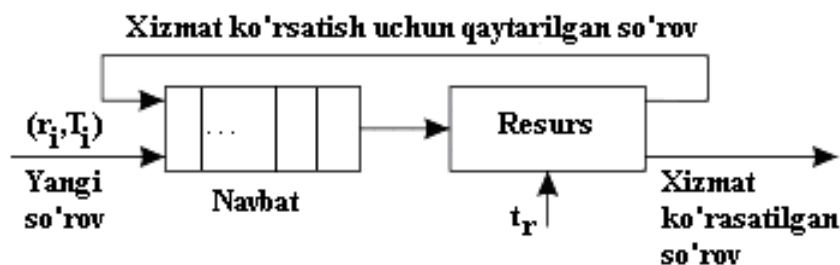
Tayanch DISKRETLASH sinfi va uning vorislari sifatida yuqorida qayd qilingan diskretlash usullariga mos ravishda DISKRET\_KV va DISKRET\_KUB sinflari yaratilsin. Voris sinflarda Diskretlash() virtual funksiyalari aniqlanib, ular diskretlangan sohaning tugun nuqtalari koordinatalarini qaytarsin.

15. Multiprogrammali operatsion sistemalarda jarayonlar o‘rtasida resurslarni taqsimlash masalasi quyidagicha tavsiflanadi:

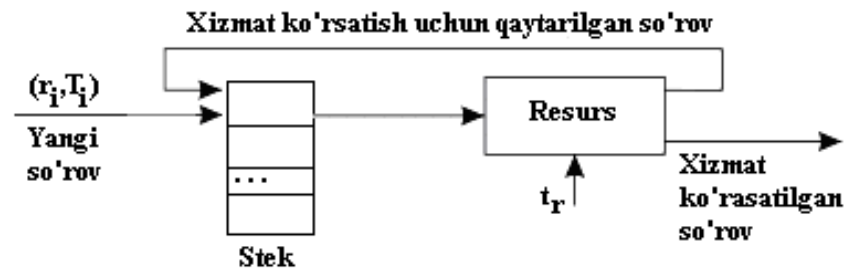
Butun  $n$  son va  $n$  ta jarayon (so‘rovlar) berilgan. Har bir  $i$  ( $i = \overline{1, n}$ )-jarayon  $(r_i, T_i)$  juftlik bilan beriladi. Bu yerda  $r_i$ - jarayon tomonidan talab qilinadigan resurs vaqti,  $T_i$  jarayonning o‘zidan oldingi  $T_{i-1}$  jarayondan qancha vaqt intervalidan keyin navbatga kelganligini bildiradi ( $T_0 = 0$ ).

Har bir jarayonga resursning  $t_r$  vaqt kvanti ajratiladi. Agar jarayon uchun resurs talabi to‘liq bajarilsa u yo‘qoladi, aks holda u navbat oxiriga qaytib keladi. Jarayonga resurs ajratishning quyidagi usullari mavjud:

a) resurslarni taqsimlashning FCFS usuli. Navbatdagi jarayonlarga “*Birinchchi kelganga birinchi xizmat*” tamoyili bo‘yicha resurs ajratiladi (rasmga qarang);



b) resurslarni taqsimlashning LCFS usuli. Navbatdagi jarayonlarga “*Oxirgi kelganga birinchi xizmat*” (stek) tamoyili bo‘yicha resurs ajratiladi (rasmga qarang).



Jarayonlar o'rtasida resurslarni taqsimlashni amalga oshiruvchi RESURS sinfi va uning vorislari sifatida RESURS\_FCFS (FCFS usuli) va RESURS\_LCFS (LCFS usuli) sinflari aniqlansin. Barcha sinflarda berilgan  $T$  vaqt uchun  $[0...T]$  intervalning har bir vaqt birligida jarayonlarga resurs ajratilishini (yoki navbatda kutishini) chop etuvchi virtual Resurs\_Taqsimoti() funksiya-a'zo aniqlansin.

16. Bir o'lchamli  $y = f(x)$  funksiyaning  $x_0, x_1, \dots, x_n$  nuqtalardagi  $f(x_0), f(x_1), \dots, f(x_n)$  qiymatlari jadval ko'rinishida berilgan.  $f(x)$  funksiyaning  $x^* \notin \{x_0, x_1, \dots, x_n\}$  nuqtadagi  $y^*$  qiymatini Lagranj yoki Nyuton interpolatsion formulasi bilan hisoblash mumkin:

1) Lagranj interpolatsion formulasi -  $y^* = L_n(x^*) = \sum_{k=0}^n \frac{\prod_{j \neq k} (x^* - x_j)}{\prod_{j \neq k} (x_k - x_j)} f(x_k);$

2) Nyuton interpolatsion formulasi -  $y^* = P_n(x^*) = f(x_0) + (x^* - x_0)f(x_0, x_1) + (x^* - x_0)(x^* - x_1)f(x_0, x_1, x_2) + \dots + (x^* - x_0)(x^* - x_1) \dots (x^* - x_{n-1})f(x_0, x_1, \dots, x_n),$

bu yerda  $f(x_j, x_{j+1}, \dots, x_{j+k}) = \sum_{i=j}^{j+k} \frac{f(x_i)}{\prod_{l=j, l \neq i}^{j+k} (x_i - x_l)}.$

Jadval ko'rinishidagi funksiya berilganlarini o'qish, saqlash va chop etish bilan bog'liq amallarni o'z ichiga oluvchi tayanch JADVAL\_FUN sinfi aniqlansin. Uning vorislari sifatida LAGRANJ va NYUTON sinflari yaratilsin va ularda berilgan  $x^*$  bo'yicha  $y^*$  qiymatni hisoblovchi Interpolyatsiya() virtual funksiyalar aniqlansin.

## 27. Grafika

C++ Builder muhitida grafika chizish bo'yicha ko'rsatmalar va namunaviy masala 3-ildovada keltirilgan.

### Amaliy topshiriqlar

- Quyidagi funksiyalar grafiklari  $x \in (-3, 3)$  oraliq uchun qurilsin:
  - $y = 3x^2;$
  - $y = 6x^2 + 3x;$
  - $y = x^3 - 2x^2 + 3.$
- Quyidagi funksiyalarning aniqlanish sohasi tekshirilsin va grafigi qurilsin:
  - $y = \frac{1}{x};$
  - $y = x + \frac{3}{x} - 2;$
  - $y = \frac{1}{x^2 + 3x + 1};$

$$\text{e) } y = 3 - \frac{2}{x} - \frac{3}{x^2}; \quad \text{g) } y = \frac{x^2}{x^2 - 3x + 1}; \quad \text{f) } y = \frac{1}{x^2} + 2x + 1.$$

3. Ekranda gorizontal bo'yicha o'zgarmas tezlikda soha chegaralariga urilib qaytuvchi shar hosil qilinsin.

4. Ekranda gorizontal bo'yicha o'zgarmas tezlikda chapdan o'ngga siklik harakat qiluvchi satr (reklama) hosil qilinsin.

5. Parametrik tenglamalar bilan aniqlangan chiziqlar chizilsin:

a) markazi koordinata boshida bo'lgan  $r$  radiusli aylana:

$$x = r \cdot \cos(t), y = r \cdot \sin(t), t \in (0, 2\pi).$$

b) katta va kichik yarim o'qlari koordinata o'qlariga parallel va radiuslari mos ravishda  $r_1$  va  $r_2$  bo'lgan ellips:  $x = r_1 \cdot \cos(t), y = r_2 \cdot \sin(t), t \in (0, 2\pi).$

d) paskal chig'anoq'i:

$$x = a \cdot \cos^2(t) + b \cdot \cos(t), y = a \cdot \cos(t) \cdot \sin(t) + b \cdot \sin(t),$$

$$a > 0, b > 0, t \in (0, 2\pi), b \geq 2a, a < b < 2a, a > b$$

hollarda ko'rilsin.

e) kardioda:  $x = a \cdot \cos(t + \cos(t)), y = a \cdot \sin(t + \cos(t)), a > 0, t \in (0, 2\pi)$

6. Funksiya grafigi chizilsin:

$$\left| y + \frac{1}{y} \right| = \left| x + \frac{1}{x} \right|, |x| < 2.$$

7. Koordinatalari quyidagi tengsizlik va tengsizliklar sistemasini qanoatlantiruvchi nuqtalar ekranda yoritilsin.

$$\text{a) } |y| + 2|x| \leq x^2 + 1; \quad \text{b) } x^2 + y^2 \leq 2(|x| + |y|);$$

$$\text{d) } 4 \leq x^2 + y^2 \leq 2(|x| + |y|); \quad \text{e) } 2y - x^2 \leq 4, x^2 + y^2 \geq 0.$$

8. Mart oyi kunlarining haroratlarini ko'rsatuvchi  $t_1, t_2, \dots, t_{31}$  butun sonlar berilgan. Harorat grafigini quring. Nol haroratga mos keluvchi gorizontal chiziqdan yuqorida va pastda to'g'ri chiziq kesmalari har xil rangga bo'yalsin.

9. Butun  $x, y, r, m, a, b$  sonlar berilgan. Markazi  $(x, y)$  nuqtada bo'lgan  $r$  radiusli aylana va yuqori chap uchi  $(m, n)$  nuqtada, bo'yi  $a$  va eni  $b$  bo'lgan to'g'ri to'rtburchak chizilsin. Aylana va to'rtburchak markazlarini tutashtiruvchi kesma yasang.

10. Butun  $n$  va  $r$  sonlari berilgan.  $r$  radiusli aylanaga ichki chizilgan ko'pburchak uchlari bo'lgan  $n$ -ta nuqta qurilsin. Har bir nuqta qolgan  $n-1$  nuqtalar bilan tutashtirilsin. Nuqtalarning koordinatasi quyidagi formulalar bilan berilgan:

$$x_t = r \cdot \cos\left(\frac{2\pi t}{n}\right), y_t = r \cdot \sin\left(\frac{2\pi t}{n}\right) (t = 1, 2, \dots, n).$$

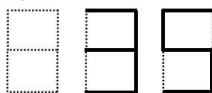
Tutashtiruvchi chiziqlarni takroran o'tkazmaslik uchun  $t$  nomerli nuqtani faqat  $t < j$  shartni qanoatlantiruvchi  $j$  nuqtalar bilan tutashtirish kerak.

11. Butun  $n$  va  $r$  sonlari berilgan. Tomonlarning uzunligi  $r$  bo'lgan kvadrat yasalsin. Kvadratning har bir uchida bittadan va har tomonida  $n-1$  tadan nuqtalar joylashtirilsin. Tomonlardagi o'zaro qo'shni nuqtalar orasidagi



masofa bir xil bo'lib,  $r/n$  soniga teng. Bu usul bilan hammasi bo'lib  $4n$  ta nuqta quriladi va ularni  $1, \dots, 4n$  sonlari bilan nomerlash mumkin (nomerlash kvadratning yuqori chap uchidan boshlanib, soat millari yo'nalishida amalga oshiriladi). Tartib nomeri  $t$  bo'lgan har bir nuqtani  $j > t$  shartni qanoatlantiruvchi va  $j-1$  ayirmasi  $4n$  sonidan kichik bo'lgan Fibonachchi sonlarini beradigan  $j$  nomerli nuqtalar bilan tutashtirilsin.

12. Ekran tekisligida quyidagi shartlar ostida aylanuvchi kesma yasalsin:
  - a) kesma o'rtasiga nisbatan;
  - b) birorta uchiga nisbatan;
  - d) kesmani  $1/3$  nisbatda bo'luvchi nuqta atrofida.
13. Ikkita ko'rsatkich-kesmalarni (soat millarini) qo'zg'almas nuqta atrofida aylanishi amalga oshirilsin. Bunda kesmalar birining (kattasining) bir marta to'liq aylanib chiqishiga ikkinchisining  $1/12$  aylanishiga mos keladi.
14. Ekranda ishlayotgan elektron soat tasviri hosil qilinsin. Soatdagi raqamlar ko'rinishi oddiy elektron soatlaridagi kabi yetti segmentli (bo'lakli) qolipga mos kelishi kerak (rasmga qarang).



15. Ekran tekisligida quyidagi shartlar bo'yicha soat millari harakati bo'yicha (yoki aksincha) aylanuvi to'g'ri burchakli uchburchak tasvirlansin:
  - a) markazi atrofida;
  - b) uchlaridan biri atrofida.
16. Ekranda aylana o'lchami va uning joylashuvi boshqarilsin. Boshlang'ich holatda aylana markazi ekran markazida joylashsin va radiusi  $r$  deb hisoblansin. Boshqarish klaviaturaning quyidagi tugmalari yordamida amalga oshirilsin. Agar '+' tugmasi bosilsa, aylana radiusi 5 pikselga kattalashsin, agar '-' tugmasi bosilsa, aylana radiusi besh pikselga kichraysin. Klaviaturaning yo'nalish tugmalari bosilsa, aylana mos yo'nalishda 5 pikselga ko'chsin.
17.  $h$  balandlikdan  $v$  boshlang'ich tezlik bilan gorizontaal yo'nalish bo'yicha otilgan jismning Yer sharining tortish kuchi ta'siri ostidagi harakati tasvirlansin. Havo qarshiligi hisobga olinmasin.
18. Yadro atrofida  $k$  ta elektronning berilgan elliptik orbita bo'ylab o'zgarmas tezlikdagi harakati tasvirlansin. Elektronlar harakati koordinata o'qlarini  $t$  radianga burish asosida amalga oshiriladi:

$$(x, y) \rightarrow (x \cdot \cos t + y \cdot \sin t, -x \cdot \sin t + y \cdot \cos t)$$

19. Ko'rinmas aylana bo'ylab harakat qiluvchi to'g'ri chiziq kesmasining tasviri hosil qilinsin.
20. Ekranda gorizontaal yo'nalishda o'zgarmas  $v$  tezlikda dumalab ketayotgan g'ildirak tasvirlansin.
21. Ekranda berilgan fokusli, qavariq linza uchun sham tasvirining oraliq masofaga bog'liq ravishda kattalashuvchi (kichrayishi) ko'rsatilsin. Biror obyekt tasvirini kattalashtirish (kichraytirish) quyidagi formula yordamida koordinata o'qlarini cho'zish (siqish) orqali amalga oshiriladi:

$$(x, y) \rightarrow (s_x \cdot x + s_y \cdot y),$$

bu yerda  $s_x, s_y$  - mos ravishda OX va OY o'qlar bo'yicha cho'zish (siqish) kattaligi.

22. Ektranda chuqurchalari bo'lmagan bilyard taxtasi ustidagi shar harakati tasvirlansin.
23. Post mashinasini ishlashi, yani karetkaning lenta bo'ylab harakati ko'rsatilsin.
24. "Hayot" o'yini tirik kataklar gipotetik koloniyasi hayotining modelidir. Unda kataklar quyida keltirilgan qoida asosida tirik qoladi, ko'payadi yoki o'ladi (mos rasmlarga qaralsin):
- a) katak tirik qoladi agar u o'z atrofida faqat ikki yoki uchta qo'shnilar ega bo'lsa (mumkin bo'lgan sakkiztadan);
  - b) katak o'ladi, agar uning atrofida faqat qo'shnilardan bittasi yoki umuman bo'lmasa;
  - d) agar katak atrofida to'rtta yoki undan ko'p qo'shnilar bo'lsa, katak o'ladi;
  - e) bo'sh joy atrofida uchta katak bo'lsa, bu joyda katak paydo bo'ladi.

Boshlang'ich holatga ko'ra kataklar hayoti tasvirlansin.

X		
	X	X

a)

	X	

b)

X	X	
	X	X
X		

d)

X		X
	X	

e)

25. "Samolyotni urish" o'yini. Ekraning yuqori qismida gorizontall uchib borayotgan samolyotini urib tushirish kerak. Urish qurilmasi ekran pastida joylashgan, u oldinga va orqaga harakat qilishi mumkin. Otilgan o'qning harakatida Yerning tortish kuchi inobatga olinsin.
26. "Quyvon va bo'ri" masalasi. Chegaralangan to'rtburchakli sohada bo'ri va quyvonning boshlang'ich joylari mos ravishda  $(x_b, y_b)$  va  $(x_q, y_q)$  nuqtalar bilan, ularning vaqt birligidagi ko'chishlari  $s_b$  va  $s_q$  bilan berilgan.

Soha bo'ylab bo'rining quyvonni quvlashdagi izi chizilsin.

Har bir vaqt birligida bo'ri va quyvonning harakati quyidagicha amalga oshiriladi:

- bo'ri quyvon tomonga  $s_b$  masofaga ko'chadi;
- quyvon bo'ridan uzoqlashadigan tomonga  $s_q$  masofaga ko'chadi.

Bo'ri va quyvon harakatida soha chegarasini (to'siqlarni) inobatga olish kerak bo'ladi, ya'ni ular to'siqlardan "qochishi" zarur bo'ladi.

Quvlash jarayoni bo'rining navbatdagi ko'chishida quyvon koordinatasini kesib o'tganda yoki oldindan ko'rsatilgan T vaqtdan keyin tugaydi.

### **Мустақил таълимни ташкил этишнинг шакли ва мазмуни**

Талаба мустақил таълимнинг асосий мақсади – ўқитувчининг раҳбарлиги ва назоратида муайян ўқув ишларини мустақил равишда бажариш учун билим ва кўникмаларини шакллантириш ва ривожлантириш.

Мустақил ишларни бажариш жараёнида талабалар қуйидаги ишларни бажаради:

- дарслик ва ўқув қўлланмалар асосида фан мавзулари бўйича назарий тайёргарлик кўриш, амалий ва лаборатория машғулотларига тайёрланиш;
- тарқатма материаллар бўйича маърузаларни чуқур ўзлаштириш;
- фан мазмунида кўрсатилмаган дастурлаш тиллари ва муҳитлари билан танишиш ва қиёсий таҳлил қилиш;

- масофавий таълим орқали дастурлаш билан турдош фанлар бўйича ўқув курсларида қатнашиш ва мос сертификатларга эга бўлиш тавсия қилинади.

Талаба мустақил ишини ташкил этишда қуйидаги шакллардан фойдаланади:

- берилган мавзулар бўйича ахборот (реферат) тайёрлаш;
- назарий билимларни амалиётда қўллаш;
- макет, модел ва намуналар яратиш;
- илмий мақола, анжуманга маъруза тайёрлаш ва ҳ.к.

### **Тавсия этилаётган мустақил ишларнинг мавзулари**

1. C ва C++ тили синтаксислари.
2. C++ ва бошқа тилларда (Pascal, C#) модулли дастурлаш.
3. Стандарт кутубхона. Оқим синфлари.
4. Алгоритмлар кутубхонаси. Сонли ҳисоблаш функцияларининг кутубхонаси.
5. .NET технологияси, C# тилини хусусиятлари.
6. 8,16,32 разрядли процессорлар.
7. Турли муҳитларда яратилган дастур объектларини боғлаш.
8. C++ тили учун консол режимидаги дастурлаш муҳитлари.
9. Динамик структураларни қайта ишлаш алгоритмлари.
10. Саралаш ва излашнинг самарали алгоритмлари.
11. C++ тили стандартлари.
12. C++ тили асосида яратилган тиллар (C# ва Java тиллари).

### **ГЛОССАРИЙ:**

**Ахборот** – сўзи латинча «informatio» сўзидан келиб чиққан бўлиб «тушунтириш, таништириш, баён этиш» - деган маъноларни англатади.

**Ахборот технологиялари** – ахборотни йиғиш, сақлаш, узатиш, ўзгартириш, қайта ишлаш усул ва воситалари йиғиндисидан иборат.

**Ахборотлаштириш** - ахборот ресурслари, ахборот технологиялари ва ахборот тизимларидан фойдаланилган ҳолда юридик ва жисмоний шахсларнинг ахборотга бўлган эҳтиёжларини қондириш учун шарт-шароитлар яратишнинг ташкилий ижтимоий-иқтисодий ва илмий-техникавий жараёни.

**Ахборот ресурслари** – инсонларнинг у ёки бу ахборотларга бўлган эҳтиёжларини қондириш учун тўпланган барча ахборотлар тушунилади.

**Ахборот тизими** – ахборот ресурслари, ахборот технологиялари ва алоқа воситаларининг ахборотни тўплаш, сақлаш, излаш, унга ишлов бериш ва ундан фойдаланиш имконини берадиган ташкилий жиҳатдан тартибга солинган мажмуи.

**Виртуал аудитория** – масофали ўқитиш шароитида ўқувчилар гуруҳи ўқув - тарбия жараёнини ташкил этиш.

**Виртуал университет** – ахборот – таълим муҳити, бошқариш блоклари ва виртуал таълим муассасасини бошқариш йиғиндиси.

**Виртуал сервер** – бу провайдер билан келишган ҳолда компьютернинг имконияти даражасига қараб алоҳида сервер кўринишда ишлашга мўлжалланган қўшимча плата (электрон схемалар) ўрнатилган компьютер ҳисобланади.

**Веб саҳифа** - оддий хат ёки китоб саҳифасидан фарқли нарсаси тушунилади. Веб саҳифа - Интернет - провайдерга хизмат қилувчи ва қандайдир фирма, ташкилот ёки шахсга тегишли хизмат қилувчининг узоқлаштирилган компютерида қандайдир ном билан бирлаштирилиб, жойлашган ахборотлар тўпламидир.

**Веб сайт** – бир нечта саҳифалар туркуми сайт (site) деб аталади. Веб сайт саҳифалари ўлчамлари шахсий компьютернинг экрани ўлчамларига автоматик равишда мослаштирилади.

**Веб портал** - сўзи инглизча сўз бўлиб, умумий маънода бирор нарсани бирлаштириш, бириктириш нуқтаси маъноларида қўлланилади. Интернет технологиясида эса турли хил универсал сервисларни бирлаштирувчи йирик сайт маъносида қўлланилади.

**Глобал тармоқлар** – турли мамлакатлар ёки қитъаларда, ер юзининг барча жойларида жойлашган тармоқ фойдаланувчиларини бирлаштиради.

**Гипермедиа** – матндан ташқари бошқа шаклдаги маълумотларни ҳам тавсия қилувчи ҳужжатлар ҳисобланади.

**Гиперматн** - Гиперматн ихтиёрий жойда жойлашган ўзаро алоқадаги ҳужжатлар орасидаги алоқа. Ажратилган жумла ёки сўзни "сичқонча" билан босилганда фойдаланувчи тезгина ушбу мавзу ёритилган файлларга ўтиши мумкин.

**Локал (маҳаллий) тармоқлар** (LAN - Local Area Network)-бир хонадаги, бинодаги, унча катта бўлмаган ҳудуддаги компьютер тармоқлари.

**Регионал (минтақавий) тармоқлар** - бир-биридан анча узоқда жойлашган компьютерлар ва маҳаллий тармоқларни ўзаро боғлайди.

**Масофали ўқитиш** – ахборот-коммуникацион технология (компьютерлар, телекоммуникация, мультимедиа) воситалари ва илмий асосланган ўқитиш усуллари қўллаб таълим (кундузги, сирки, экстернат) олиш шаклидир.

**Масофаи таълим** (distant education) – ахборот технологиясини фойдаланган ҳолда масофадан туриб таълим муҳити ёрдамида ўқув ахборотларини алмашишни таъминлайдиган ва ўқув жараёнини олиб бориш ҳамда бошқариш тизимини амалга оширадиган билим ва кўникмаларни эгаллаш жараёни.

**Электрон почта** (E mail)- Жўнатиш учун мўлжалланган электрон хат, яъни матнли файл фойдаланувчининг манзили ва хат мазмунидан ташкил топади.

**Электрон дарслик** - компьютер технологиясига асосланган ўқув услубини қўллашга, мустақил таълим олишга ҳамда фанга оид ўқув материаллар, илмий маълумотларнинг ҳар томонлама самарадор ўзлаштирилишига мўлжалланган.

**INTERNET** - International Network (халқаро компьютер тармоғи) - бутун дунё компьютер тармоғи. Internet – WWW гиперматнларга асосланиб ахборот ташкил, узатиш ва қабул қилиш тизимини амалга оширадиган халқаро компьютер тармоқидир.

**WWW** - World Wide Web ("Умумжаҳон ўргимчак тўри") - гипермуҳитга асосланиб Интернетда ташкил қилинган ахборотлар тизими.

**URL** (Uniform Resource Locator - ашёларни унификация қилинган кўрсатувчиси) - Web-тугуннинг манзили. URL кўрсаткичи одатда ҳужжатнинг транспорт баённомасини (масалан, HTTP ёки FTP) ва у жойлашган host - компьютернинг номини ифодалайди. Бундан ташқари URL кўрсаткичлари ўзларида ушбу компьютердаги ҳужжатга кириш маршрутини ҳам сақлаши мумкин. Ушбу маршрутлар URL сатрининг охирида кўрсатилади.

**Интернет портали** - деганда интернетдан фойдаланувчиларга бошқа сайтлардан ахборотлар олишга имкон берувчи, ўзида кўплаб ўқув-услугий, илмий материалларни, меъёрий-ҳуқуқий ҳужжатларни муҳассамлаштириш имконияти мавжуд веб ресурсдир.

**Intranet** (Intranet -интратармоқ) - дастурий маҳсулотлари ва Интернет технологияларини фойдаланиладиган бирор ташкилотнинг маҳаллий корпоратив тармоқидир, масалан, Web-сервер Интратармоқлар брендмауерлар деб аталувчи махсус дастурлар ёрдамида Интернетнинг ташқи фойдаланувчиларидан ажратилиб қўйилиши ёки ташқаридан алоқа қилаолмайдиган автоном тармоқ сифатида ишлатилиши мумкин.

**Тармоқ абонентлари** – тармоқда ахборотларни юзага келтирувчи ёки истеъмол қилувчи объектлардир. Уларга алоҳида компьютерлар, компьютер комплекслари, саноат иншоотлари, дастур орқали бошқариладиган станоклар ва бошқалар мисол бўлиши мумкин. Ҳар қандай абонент тармоғи станцияга уланган бўлади.

**Станция** – ахборот узатиш ва қабул қилиш билан боғлиқ вазифаларни бажарувчи техник жиҳозлар йиғиндисидир.

**Узел** – тармоқнинг узатиш воситасига уланган ҳар қандай

қурилмалар.

**Сервер** - бу тармоқдаги компьютерлар ва уларга хизмат кўрсатадиган ташкилотнинг компьютерларидир.

**Сервис–провайдер** – компьютер тармоқлари хизматини ташкил қилувчи ташкилотлар ҳисобланади.

**HTML формати** (Hyper Text Markup Language – гиперматнни белгилаш тили) – Web саҳифаларни яратишда қўлланиладиган тил ҳисобланади.

**Мультимедиа воситалари** - информатиканинг дастурий ва техникавий воситалари асосида аудио, видео, матн, графика ва анимация (объектларнинг фазодаги ҳаракати) эффектлари асосида ўқув материалларини ўқувчиларга етказиб беришнинг мужассамланган ҳолдаги кўринишидир.

**Замонавий ахборот технологиялари** – шахсий компьютерлар ва телекоммуникация воситаларидан фойдаланган ҳолда фойдаланувчи ишининг дўстона интерфейсли ахборот технологиясидир.

**Иловалар**

**Фан дастури**

ЎЗБЕКИСТОН РЕСПУБЛИКАСИ  
ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ ВАЗИРЛИГИ

Рўйхатга олинди:  
№ БД – 5130100 – 2.01  
2017 йил “18” 08



ДАСТУРЛАШ АСОСЛАРИ  
ФАН ДАСТУРИ

Билим соҳаси: 100 000 – Гуманитар соҳа  
Таълим соҳаси: 130000 – Математика  
Таълим йўналиши: 5130100 – Математика

Тошкент – 2017



---

Ўзбекистон Республикаси Олий ва ўрта махсус таълим вазирлигининг 2017 йил "14" 08 даги "605"-сонли буйруғининг 2-қисми билан фан дастури рўйхати тасдиқланган.

Фан дастури Олий ва ўрта махсус, касб-хунар таълими йўналишлари бўйича Ўқув-услубий бирлашмалар фаолиятини Мувофиқлаштирувчи Кенгашининг 2017 йил "18" 08 даги 4 - сонли баённомаси билан маъқулланган.

Фан дастури Мирзо Улугбек номидаги Ўзбекистон Миллий университетида ишлаб чиқилди.

**Тузувчи:**

Полатов А.М. - физика-математика фанлари доктори, ЎзМУ проф.в.б.;

**Такрибчи:**

Гулямов Ш.М. - техника фанлар доктори, ТДТУ профессори;

Фан дастури Мирзо Улугбек номидаги Ўзбекистон Миллий университети Услубий кенгашида кўриб чиқилган ва тасвир килинган (2017 йил "14" 08 даги 4 - сонли баённома).

## **I. Ўқув фанининг долзарблиги ва олий касбий таълимдаги ўрни**

“Дастурлаш асослари” фанининг бош мақсади талабаларга қўйилган масалани ечадиган компьютер дастурини тузиш асосларини ўргатишдир. Шу мақсадда дастурлаш тиллари ва муҳитлари ҳақида таянч тушунчалар берилади ва бу тиллардан фойдаланишга ўргатилади.

Фан назарий ва амалий қисмлардан иборат. Назарий қисм информатика ва ҳисоблаш техникаси, алгоритмлар, C/C++ дастурлаш тили, C++Builder объектга йўналтирилган дастурлаш муҳитларида ишлаш бўйича қўрсатмалар бўлимларидан ташкил топган.

Дастурда компьютерда дастурлашга киришнинг назарий асоси бўлган алгоритмларга алоҳида эътибор қаратишган. Бу ерда алгоритмларни тавсифлаш ва кейинчалик компьютерда амалга ошириш учун зарур бўлган бир қатор математик тушунчалар - такрорлаш, ёрдамчи алгоритм, рекурсия, хотира, массив, индекс, функция, параметр ва ҳ.к. киритилиб, турли хил синф масалаларининг алгоритмлари тузилади.

Дастурлаш тили - тузилган алгоритмни компьютер амалга ошириши учун воситадир. Бу ўринда турли мураккабликтаги синтаксис ва семантикага эга бўлган тиллардан фойдаланиш мумкин.

“Дастурлаш асослари” фани йўналишининг ўқув режасидаги “Эхтимоллар назарияси”, “Солиқ усуллар”, “Дискрет математика ва математик логика” фанлари билан узвий боғлиқ. Фан мазмунини йўналишининг ўқув режасидаги “Математик статистика”, “Илмий ҳисоблашлар”, “Механика”, “Оддий дифференциал тенгламалар”, “Хусусий ҳосиллали дифференциал тенгламалар” фанларини ўзлаштиришда таянч ҳисобланади.

“Дастурлаш асослари” фани умумкасбий фан ҳисобланади ва ўқув йилининг 1-2 семестрларида ўқитилади. Фанни ўқитиш маъруза, амалий машғулот ва мустақил таълим шаклида олиб борилади.

Мазкур дастурга кўра ушбу фан доирасида қўлаб модел масалалар ўргатиладики, бу мазкур фанни чуқур ўрганган ҳар бир бакалавр олган билим ва қўникмаларини ишлаб-чиқаришда, илмий-тадқиқот ишларида, шунингдек, таълим тизимида самарали фойдаланиши имконини беради.

## **II. Фаннинг мақсад ва вазифалари**

**Фанни ўқитишдан мақсад** – “Математика” йўналишининг бакалавр босқичи талабаларига дастурлаш асосларини етарли даражада ўқитиш, шу билимларга таянган ҳолда компьютер ёрдамида моделлаштиришга келадиган талбиқий масалаларнинг дастур таъминотини амалга оширишга ўргатиш ва ихтисослик фанларини ўзлаштиришда таянч билимларга эга бўлиш.

**Фаннинг вазифалари** – масала ечишнинг алгоритмик асосларини ўрганиш, компьютер ишлашининг тамоили, дастурлаш тилларини синфлаш, компьютерда берилганлар ва буйруқларни тасвирланиши, C++ тилида дастурлаш, объектга йўналтирилган дастурлаш технологиялари, визуал дастурлаш муҳитида ишлаш бу фанининг асосий вазифалари ҳисобланади.

“Дастурлаш асослари” фанини ўзлаштириш жараёнида амалга ошириладиган масалалар доирасида бакалавр ахборот, уни сақлаш усуллари, қайта ишлаш ва узатиш, ҳисоблаш тизимларининг математик ва дастурий таъминоти, уларни фан соҳаларида, ишлаб чиқариш ва таълимда қўллаш хусусиятлари, компьютерни дастурий таъминоти, дастур турлари ва хусусиятлари, структурали, объектга йўналтирилган ва умумлашган дастурлаш, дастурни оптимallasштириш ва умумallasштириш, дастурлашда модулли таъминотларини қўллаш, компьютер технологиялари ютуқларини замонавий ҳисоблаш тизимларининг математик ва дастурий таъминотида қўллаш, дастурлашнинг тараққиётининг аънавалари ҳақида тасаввурга эга бўлиш, юқори даражадаги дастурлаш тилларини, дастурий таъминотни, дастурлаш технологияларини, табиқий ва ҳисоблаш математикаси масалаларини ечиш алгоритмларини, модулли таҳлил ва модулли дастурлаш асосларини, объектга йўналтирилган ва умумлашган дастурлаш усулларини, самарали дастур ва дастурлар комплексини яратиш усулларини билиш ва улардан фойдаланиш олиш, табиқий масалаларни ечиш алгоритминини тузиш, математик (компьютер) моделини қуриш ва унинг дастурий таъминотини яратиш, структурали, объектга йўналтирилган ва умумлашган дастурлаш парадигмаларини қўллаш асосида иловаларни ярата олиш, дастурлашда, ҳисоблаш техникаси ва дастурий таъминот имкониятларидан самарали фойдаланиш, муаммога ва объектга йўналтирилган тиллардан фойдаланиш, яратилган иловаларни баҳолаш қўникмаларига эга бўлиш керак.

### III. Асосий назарий қисм (маъруза машғулоти)

C++ тили синтаксиси ва унинг лексик асоси. C++ тили дастурининг тузилиши ва шакли. Берилганлар турлари. C++ тилининг таъин турлари. Ўзгарувчилар ва ифодалар. Амаллар: инкремент, декремент, sizeof, логикий, разрядли, таққослаш. Ўқиш-ёзиш оқимлари (cin, cout).

Операторлар. Шарт операторлари. Такрорлаш операторлари. Бошқарувни узатиш операторлари.

Статик массивлар. Функциялар эълон қилиш ва аниқлаш. Локал ва глобал ўзгарувчилар. Рекурсив функциялар. Стандарт кутубхона функциялари. Кўрсаткичлар ва адрес олувчи ўзгарувчилар. Динамик массивлар. Функция ва массивлар.

Сатр ва улар устида амаллар. Тузилмалар ва бирлашмалар. Динамик тузилмалар.

Файл тушунчаси. Матн ва бинар файллар. Файл ва сатр оқимлари. Файлдан ўқиш-ёзиш функциялари. Файл кўрсаткичининг бошқариш функциялари.

C++ тилида синфлар. Синфини ва объектларни таърифлаш. Синф майдонлари ва методлари. Конструктор ва Деструкторлар. Операторларни қайта юклаш. Ворислик.

#### **IV. Амалий машгулотларни ташкил этиш бўйича кўрсатма ва тавсиялар**

Амалий машгулотлар ўтказилишидан мақсад дастурлаш бўйича олинган назарий билимларни амалда мустахкамлаш ва турли ҳолатдаги масалаларни ечишга қўлашдан иборат. Амалий машгулотларни бир қисми аудиторияда доскада ечилиши билан ўтказилса, унинг катта қисми бевосита компьютерда амалга оширилиши керак.

#### **Амалий машгулотлар учун тавсия этилаётган тахминий мавзулар рўйхати**

1. Санок системалари. Бир санок системасидан иккинчисига ўтиш.
2. Берилганларни компьютер хотирасида тасвирланиши. Кодлаш.
3. Масалани ечиш алгоритмини тузиш ва унинг кўринишлари.
4. Бутун сонли арифметика масалалари.
5. Ичма-ич жойлашган такрорланувчи жараёнлар, итерацион жараёнлар.
6. Кетма-кетликларни тартиблаш, оддий саралаш масалалари.
7. C++ тили синтаксиси. C++ тилида дастур тузилиши.
8. Visual C++ муҳитида ишлаш.
9. Ўзгарувчилар, амаллар, ифодалар билан ишлаш.
10. Ўқиш-ёзиш оқимларида (cin, cout) киритиш-чиқариш усулларида фойдаланиш.
11. Шарт операторлари билан ишлаш.
12. Такрорлаш операторлари билан ишлаш.
13. Бошқарувни узатиш операторлари билан ишлаш.
14. Статик массивлар билан ишлаш.
15. Функцияларни эълони ва аниқлаш. Оддий функциялар тузиш.
16. Функция параметрлари ва қайтарувчи қийматлари билан ишлаш.
17. Рекурсив функциялар билан ишлаш.
18. Стандарт кутубхона функцияларидан фойдаланиш.
19. Кўрсаткичлар билан ишлаш.
20. Динамик массивлар билан ишлаш.
21. ASCII сатрлар ва улар устида амаллар.
22. string туридаги сатрлар ва улар устида амаллар.
23. Тузилмалар билан ишлаш. Бирлашмалар билан ишлаш.
24. Матн файллар ва улар устида амаллар.
25. Бинар файллар ва улар устида амаллар.
26. Файл структуралар билан ишлаш, саралаш ва кидириш алгоритмлари, ифодаларини ҳисоблаш.
27. Динамик тузилмалар билан ишлаш.
28. Синф объектларини яратиш ва улар устида амаллар.
29. Операторларни қайта юклаш.

*Илоҳ:* Амалий машгулот соатлари ҳажмларидан келиб чиққан ҳолда ишчи дастурда мазкур мавзулар ичидан амалий машгулот мавзулари шакллантирилади.



#### **V. Мустақил таълимни ташкил этишнинг шакли ва мазмуни**

Талаба мустақил таълимнинг асосий мақсади – ўқитувчининг раҳбарлиги ва назоратида муайян ўқув ишларини мустақил равишда бажариш учун билим ва кўникмаларини шакллантириш ва ривожлантириш.

Мустақил ишларни бажариш жараёнида талабалар қуйидаги ишларни бажарадилар:

- дарслик ва ўқув қўлланмалар асосида фан мавзулари бўйича назарий тайёргарлик кўриш, амалий ва лаборатория машғудотларига тайёрланиш;
- тарқатма материаллар бўйича маърузаларни чуқур ўзлаштириш;
- фан мазмунида кўрсатилмаган дастурлаш тиллари ва муҳитлари билан танишиш ва қиёсий таҳлил қилиш;
- масофавий таълим орқали дастурлаш билан турдош фанлар бўйича ўқув курсларида қатнашиш ва мос сертификатларга эга бўлиш тавсия қилинади.

Талаба мустақил ишнинг ташкил этишда қуйидаги шакллардан фойдаланади:

- берилган мавзулар бўйича ахборот (реферат) тайёрлаш;
- назарий билимларни амалиётда қўллаш;
- макет, модел ва намуналар яратиш;
- илмий мақола, анжуманга маъруза тайёрлаш ва ҳ.к.

#### **Тавсия этиладиган мустақил ишларнинг мавзулари**

1. C ва C++ тили синтаксислари.
2. C++ ва бошқа тилларда (Pascal, C#) модулли дастурлаш.
3. Стандарт кутубхона, Оқим синфлари.
4. Алгоритмлар кутубхонаси. Сонли ҳисоблаш функцияларининг кутубхонаси.
5. .NET технологияси, C# тилини хусусиятлари.
6. 8,16,32 разрядли процессорлар.
7. Турли муҳитларда яратилган дастур объектларини боғлаш.
8. C++ тили учун консол режимидаги дастурлаш муҳитлари.
9. Динамик структураларни қайта ишлаш алгоритмлари.
10. Саралаш ва излашнинг самарали алгоритмлари.
11. C++ тили стандартлари.
12. C++ тили асосида яратилган тиллар (C# ва Java тиллари).

*Изоҳ:* Мустақил таълим соатлари ҳажмларидан келиб чиққан ҳолда ишчи дастурда мазкур мавзулар ичидан мустақил таълим мавзулари шакллантирилади.

#### **Фойдаланиладиган адабиётлар рўйхати**

##### **Асосий адабиётлар**

1. Bjarne Stroustrup. The C++ Programming Language (3th Edition). Addison-Wesley, 1997.

2. D.S. Malik. C++ Programming: From Problem Analysis to Program Design. Fifth Edition. Course Technology, 2011.
3. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида дастурлаш асослари// Тошкент, ЎЗМУ, 2009, 196 бет.
4. Madraximov Sh.F., Ikramov A.M., Babajanov M.R. C++ tilida programmalash bo'yicha masalalar to'plami. O'quv qo'llanma // Toshkent, O'zbekiston Milliy Universiteti, "Universitet" nashriyoti, 2014. - 160 bet.

#### **Қўшимча алабийётлар**

5. Sh. Mirziyoyev Buyuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz. Toshkent "O'zbekiston" 2017. 488 b.
6. Sh. Mirziyoyev. Qonun ustuvorligi va inson manfaatlarini ta'minlash – yurt taraqqiyoti va xalq farovonligining garovi. O'zbekiston Respublikasi Konstitutsiyasi qabul qilinganligining 24 yilligiga bag'ishlangan tantanali marosimdagi ma'ruza. 2016-yil 7-dekabr. Toshkent - "O'zbekiston" - 2017. 32 b.
7. Мирзиёев Ш.М. Танкидий таҳлил, катъий тартиб-интизом ва шахсий жавобгарлик – ҳар бир раҳбар фаолиятининг кундалик қоидаси бўлиши керак. Ўзбекистон Республикаси Вазирлар Маҳкамасининг 2016 йил якунлари ва 2017 йил истиқболларига бағишланган мажлисдаги Ўзбекистон Республикаси Президентининг нутқи. // Халқ сўзи газетаси. 2017 йил 16 январь, №11.
8. Ш.Мирзиёев. Эркин ва фаровон, демократик Ўзбекистон давлатини биргаликда барпо этамиз. Ўзбекистон Республикаси Президенти лавозимига киришиш тантанали маросимига бағишланган Олий Мажлис палаталарининг қўшма мажлисидаги нутқи. Тошкент – "Ўзбекистон". 2016. 56 б.
9. ЎзР «Ахборотлаштириш тўғрисида» 2003 йил 11 декабрдаги Қонуни.
10. Bjarne Stroustrup. The C++ Programming Language (4th Edition). Addison-Wesley, 2013. 1363 page.
11. Bjarne Stroustrup. Programming: Principles and Practice using C++ (Second Edition)" Addison-Wesley, 2014. 1305 page.
12. Павловская Т.А. C++. Программирование на языке высокого уровня – СПб.: Питер. 2005.- 461 с.
13. Walter Savitch. Absolute C++, 5th edition. Addison-Wesley/Pearson, 2012. 984 page.
14. Павловская Т.С. Щупак Ю.С. C/C++. Структурное программирование. Практикум.-СПб.: Питер, 2002.-240с
15. Глушаков С.В., Коваль А.В., Смирнов С.В. Язык программирования C++: Учебный курс.- Харьков: Фолио; М.: ООО «Издательство АСТ», 2001.- 500с.
16. Культин Н.Б. C++Builder в задачах и примерах.-СПб.: БХВ-Петербург, 2005.-336с.

17. Абрамов С.А., Гнезделова Капустина Е.Н. и др. Задачи по программированию. - М.: Наука, 1983.

**Электрон манбалар**

1. <http://cppstudio.com> – C++ тилида программалаш бўйича намуналар изоҳлари билан келтирилган
2. <http://cplusplus.com> – C++ тилида мавжуд конструкциялар таърифи, ишлатиш намуналари билан келтирилган.
3. <http://www.compteacher.ru/programming> – дастурлаш бўйича видео дарсликлар мавжуд.
4. <http://www.intuit.ru> – интернет университети, дастурлаш бўйича ёзма ва видео маърузалар ўқиш, тест синувларидан ўтиш ва сертификат олиш имконияти мавжуд.
5. <http://www.ziyounet.uz> – дастурлаш асослари бўйича рефератлар топиш мумкин.

## **Ишчи фан дастури**

**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ**

**ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ ВАЗИРЛИГИ**

**ГУЛИСТОН ДАВЛАТ УНИВЕРСИТЕТИ**

**АХБОРОТ ТЕХНОЛОГИЯЛАРИ  
КАФЕДРАСИ**

**“ТАСДИҚЛАЙМАН”**

Ўқув ишлари бўйича проректор  
Н.Р. Баракаев

«\_\_» \_\_\_\_\_ 2017 й.

## **Дастурлаш асослари**

**фани бўйича**

## **ишчи ўқув дастури**

Билим соҳаси: Гуманитар  
Таълим соҳаси: 130000 – Математика  
Таълим йўналиши: 5130100-Математика

Умумий ўқув соати	– 366
Шу жумладан:	
Маъруза	– 94
Амалиёт машғулоти	– 122
Мустақил таълим соати	– 150

ГУЛИСТОН – 2017 й.



Фаннинг ишчи ўқув дастури намунавий ўқув дастури ва ўқув режасига мувофиқ ишлаб чиқилди.

**Тузувчи:** Қаландаров А.А. – ГулДУ “Ахборот технологиялари” кафедраси  
катта ўқитувчиси \_\_\_\_\_ (имзо)

**Такризчи:** Тоштемиров Д.Э. – ГулДУ “Ахборот технологиялари”  
кафедраси доценти \_\_\_\_\_ (имзо)

Фаннинг ишчи ўқув дастури “Ахборот технологиялари” кафедрасининг 2017 йил  
“\_\_\_” \_\_\_\_\_ даги \_\_\_ - сонли мажлисида кўриб чиқилиб, факультет Илмий-  
услубий Кенгашида кўриб чиқиш учун тавсия қилинди.

**Кафедра мудири:**

**доц. Абдурахимов Д.Б.**

Фаннинг ишчи ўқув дастури “Физика-математика” факультети Илмий-услубий  
Кенгашининг 2017 йил “\_\_\_” \_\_\_\_\_ даги “\_\_\_” - сонли мажлисида тасдиқланди.

Факультет Илмий-услубий  
Кенгаши раиси:

доц. Ш. Аширов

Келишилди:

Ўқув ишлари бўйича проректор

доц. Баракаев Н.

## КИРИШ

“Дастурлаш асослари” фанининг бош мақсади талабаларга қўйилган масалани ечадиган компьютер дастурини тузиш асосларини ўргатишдир. Шу мақсадда дастурлаш тиллари ва муҳитлари ҳақида таянч тушунчалар берилади ва бу тиллардан фойдаланишга ўргатилади.

Фан назарий ва амалий қисмлардан иборат. Назарий қисм информатика ва ҳисоблаш техникаси, алгоритмлар, C/C++ дастурлаш тили, Visual C++ объектга йўналтирилган дастурлаш муҳитларида ишлаш бўйича кўрсатмалар бўлимларидан ташкил топган.

Дастурда компьютерда дастурлашга киришнинг назарий асоси бўлган алгоритмларга алоҳида эътибор қаратилган. Бу ерда алгоритмларни тавсифлаш ва кейинчалик компьютерда амалга ошириш учун зарур бўлган бир қатор математик тушунчалар – такрорлаш, ёрдамчи алгоритм, рекурсия, хотира, массив, индекс, функция, параметр ва ҳ.к. киритилиб, турли хил синф масалаларининг алгоритмлари тузилади.

Дастурлаш тили – тузилган алгоритмни компьютер амалга ошириши учун воситадир. Бу ўринда турли мураккабликдаги синтаксис ва семантикага эга бўлган тиллардан фойдаланиш мумкин.

### **Ўқув фанининг мақсади ва вазифалари**

**Фанни ўқитишдан мақсад** – “Математика” йўналишининг бакалавр босқичи талабаларига дастурлаш асосларини етарли даражада ўқитиш, шу билимларга таянган ҳолда компьютер ёрдамида моделлаштиришга келадиган тадбиқий масалаларнинг дастур таъминотини амалга оширишга ўргатиш ва ихтисослик фанларини ўзлаштиришда таянч билимларга эга бўлиш.

**Фаннинг вазифалари** – масала ечишнинг алгоритмик асосларини ўрганиш, компьютер ишлашининг тамоили, дастурлаш тилларини синфлаш, компьютерда берилганлар ва буйруқларни тасвирланиши, C++ тилида дастурлаш, объектга йўналтирилган дастурлаш технологиялари, визуал дастурлаш муҳитида ишлаш бу фаннинг асосий вазифалари ҳисобланади.

### **Фан бўйича талабаларнинг билим, малака ва кўникмаларига қўйиладиган талаблар**

“Дастурлаш асослари” фани йўналишнинг ўқув режасидаги “Эхтимоллар назарияси”, “Сонли усуллар”, фанлари билан узвий боғлиқ. Фан мазмуни йўналишнинг ўқув режасидаги “Математик статистика”, “Илмий ҳисоблашлар”, “Оддий дифференциал тенгламалар”, “Хусусий ҳосилали дифференциал тенгламалар” фанларини ўзлаштиришда фойдаланилади.

### **Фаннинг ўқув режадаги бошқа фанлар билан ўзаро боғлиқлиги**

Бу фанни ўрганишда математик анализ, олий алгебра, ва бошқа фанлардаги маълумотлардан фойдаланилади. Замонавий математиканинг баъзи бўлимларига тадбиқий масалалар учраганда буни алоҳида таъкидлаб ўтилади.

## **Фаннинг ишлаб чиқаришдаги ўрни**

Мазкур дастурга кўра ушбу фан доирасида кўплаб модел масалалар ўрганиладики, бу мазкур фанни чуқур ўрганган ҳар бир бакалавр олган билим ва кўникмаларини ишлаб-чиқаришда, илмий-тадқиқот ишларида, шунингдек, таълим тизимида самарали фойдаланиши имконини беради.

## **Фанни ўқитишда замонавий ахборот ва педагогик технологиялар**

Талабаларга фаннинг баъзи мавзулар бўйича дарслар электрон воситалар ёрдамида ташкил қилинади. Талабаларнинг фанни ўзлаштиришлари учун ўқитишнинг илғор ва замонавий усулларида фойдаланиш, янги информацион-педагогик технологияларни тадбиқ этиш муҳим аҳамиятга эга. Фанни ўзлаштиришда дарслик, ўқув ва услубий қўлланмалар, маъруза матнлари, тарқатма материаллар, виртуал стендлардан фойдаланилади. Бунда Powerpoint, Maple, Mathcad каби янги математик дастурлар пакетларидан ва мавжуд электрон дарсликлар, вебсайтлардан фойдаланилади. Маъруза ва амалий дарсларида мос равишдаги илғор педагогик технологиялар ишлатилади. Бу борада замонавий педагогик технологияларининг ақлий хужум, мунозарали дарс ва бошқа усулларида фойдаланиш тавсия этилади. Ушбу фанни ўқитиш жараёнида традицион шакллардан ташқари янги педагогик технологияларнинг презентация, ақлий хужум ва бошқа усулларида фойдаланилади.

Мазкур фанни ўқитиш жараёнида таълимнинг замонавий усуллари, янги педагогик ва ахборот – коммуникация технологиялари қўлланилиши назарда тутилган. Дастурдаги барча маъруза мавзуларини ўқишда таълимнинг замонавий усулларида кенг фойдаланиш, ўқув жараёнини янги педагогик технологиялар асосида ташкил этиш самарали натижа беради. Бу борада замонавий педагогик технологиянинг “Бумеранг”, “Елпиғич”, “Ақлий хужум”, “Масофавий таълим”, “Занжир”, “Кластер” ҳамда “Муаммоли таълим” технологиясининг “Мунозарали дарс” каби усуллари қўллаш ўринлидир. Шунингдек, амалий машғулотлар жараёнида фанга тегишли бўлган компьютерлар, жадваллар, чизмалар ва слайдлардан фойдаланиш назарда тутилади.

## **Асосий қисм**

### **Фаннинг назарий машғулотлари мазмуни**

C++ тили синтаксиси ва унинг лексик асоси. C++ тили дастурининг тузилиши ва шакли. Берилганлар турлари. C++ тилининг таянч турлари. Ўзгарувчилар ва ифодалар. Амаллар: инкремент, декремент, sizeof, логикий, разрядли, таққослаш. Ўқиш-ёзиш оқимлари (cin, cout).

Операторлар. Шарт операторлари. Такрорлаш операторлари. Бошқарувни узатиш операторлари. Статик массивлар. Функциялар эълон қилиш ва аниқлаш. Main() функцияси. Локал ва глобал параметрлар. Рекурсив функциялар. Фойдаланувчи томонидан аниқланган берилганлар турлари. Номлар фазоси. Стандарт кутубхона функциялари.

Кўрсаткичлар ва адрес олувчи ўзгарувчилар. Динамик массивлар. Функция ва массивлар. Сатрлар. Сатр устида амаллар. Сатр функциялари. Тузилмалар. Бирлашмалар. Препроцессор директивалари.

Идентификаторларнинг амал доираси. Макросларни аниқлаш ва жойлаштириш. Дастурни созлаш технологиялари. Стандарт оқимлар. Берилганларни форматлаш. Оқимлар билан ишлаш. Ўқиш – ёзиш функциялари.

Файл тушунчаси. Матн ва бинар файллар. Файл ва сатр оқимлари. Форматли ўқиш ва ёзиш функциялари. Файлдан ўқиш-ёзиш функциялари. Файл кўрсаткичини бошқариш функциялари. Динамик тузилмалар. Берилганларнинг динамик тузилмалари: чизиқли рўйхатлар, стеклар, навбатлар ва бинар дарахтлар.

C++ тилида синфлар. Синфни ва объектларни тавсифлаш. Синф майдонлари ва методлари. Конструктор ва Деструкторлар. Операторларни қайта юклаш. Ворислик.

**Фандан ўтиладиган мавзулар ва улар бўйича машғулот турларига ажратилган соатларнинг тақсимооти**

№	Фаннинг бўлими ва мавзуси, маъруза мазмуни	Соатлар			
		Жами	Маъруза	Амалий Машғулот	Мустақил иш
1	Кириш. Амалий масалаларни ечиш алгоритмлари.	6	2	2	2
2	Алгоритмларни блок-схема кўринишида ифодаланиши	10	2	4	4
3	C++ тили синтаксиси ва унинг лексик асоси.	8	2	2	4
4	C++ тили дастурининг тузилиши ва шакли	6	2	2	2
5	Берилганлар турлари. C++ тилининг таянч турлари.	8	2	2	4
6	Берилганларни компьютер хотирасида тасвирланиши	6	2	2	2
7	Ўзгарувчилар ва ифодалар.	8	2	2	4
8	Амаллар: инкремент, декремент, sizeof, мантикий, разрядли, таққослаш.	10	2	4	4
9	Амалларнинг устунликлари ва бажарилиш йўналишлари.	6	2	2	2
10	Ўқиш-ёзиш оқимлари (cin, cout).	8	2	2	4
11	Операторлар.	6	2	2	2
12	Шарт операторлари	10	2	4	4
	<b>ОБ</b>				
13	Танлаш оператори	8	2	2	4
14	for такрорлаш операторлари..	8	2	2	4
15	while, do-while такрорлаш операторлари	10	2	4	4
16	Бошқарувни узатиш операторлари	8	2	2	4
17	Структурали дастурлаш	6	2	2	2
18	Бир ўлчовли статик массивлар.	8	2	2	4
19	Бир ўлчовли статик массивларустида амаллар	8	2	4	2
20	Кўп ўлчовли статик массивлар.	8	2	2	4
21	Кўп ўлчовли статик массивлар устида амаллар	8	2	4	2
22	Функциялар эълон қилиш ва аниқлаш. main()	6	2	2	2

	функцияси.				
23	Локал ва глобал ўзгарувчилар.	6	2	2	2
24	Функция параметрларида статик массивлардан фойдаланиш	7	2	2	3
	<b>ОБ</b>				
	<b>ЯБ</b>				
25	Рекурсив функциялар.	8	2	2	4
26	Фойдаланувчи томонидан аниқланган берилганлар турлари.	8	2	2	4
27	Номлар фазоси.	6	2	2	2
28	Стандарт кутубхона функциялари.	10	2	4	4
29	Кўрсаткичлар ва адрес олувчи ўзгарувчилар.	8	2	4	2
30	Динамик массивлар.	10	2	4	4
31	Функция ва массивлар.	8	2	2	4
32	Сатрлар. Сатр устида амаллар.	10	2	4	4
33	Сатр функциялари.	10	2	4	4
34	Тузилмалар.	6	2	2	2
35	Бирлашмалар.	8	2	2	4
36	Дастурни сошлаш технологиялари.	8	2	2	4
	<b>ОБ</b>				
37	Стандарт оқимлар. Оқимлар билан ишлаш. Ўқиш - ёзиш функциялари.	8	2	2	4
38	Файл тушунчаси. Матн файллар.	10	2	4	4
39	Бинар файллар	6	2	2	2
40	Форматли ўқиш ва ёзиш функциялари.	8	2	2	4
41	Динамик тузилмалар.	10	2	4	4
42	Берилганларнинг динамик тузилмалари: чизиқли рўйхатлар, стеклар,.	6	2	2	2
43	Навбатлар ва бинар дарахтлар	6	2	2	2
44	C++ тилида синфлар. Синфни ва объектларни тавсифлаш.	7	2	2	3
45	Синф майдонлари ва методлари.	6	2	2	2
46	Конструктор ва Деструкторлар	6	2	2	2
47	Операторларни қайта юклаш. Ворислик..	10	2	4	4
	<b>ОБ</b>				
	<b>ЯБ</b>				
<b>Жами</b>		<b>366</b>	<b>94</b>	<b>122</b>	<b>150</b>

## 1. Ўқув материаллари мазмуни

### 1.1. Амалий машғулотлар мазмуни

#### Амалий машғулотларини ташкил этиш бўйича кўрсатма ва тавсиялар

Ҳар бир амалий машғулот, дастлаб ишнинг мақсадини ва мавзуга оид назарий билимларни қисқача ёритишдан бошланади. Сўнг ишни бажариш учун зарур бўлган маълумотлар ва қўйилган мақсадни амалга ошириш учун талаб қилинган вазифалар аниқ белгиланиб, ишни бажариш тартиби эса қўйилган вазифалар кетма-кетлигига

асосланади. Барча ишлар олинган натижаларнинг таҳлили билан якунланади. Ҳар бир амалий машғулоти бажариш учун берилган маълумотларга таяниб, талабаларга алоҳида вариантлар таклиф этилади. Амалий машғулотларни ташкил этиш бўйича кафедра профессор-ўқитувчилари томонидан услубий кўрсатма ва тавсиялар ишлаб чиқилади. Унда талабалар асосий маъруза мавзулари бўйича олган билим ва кўникмаларни амалий масалалар компьютерда ечиш орқали янада бойитадилар. Бунга жамoa бўлиб машқ қилиш йўли билан ва мустақил ишлаш йўли билан эришилади. Мустақил ишлашда дарсликларни, ўқув қўлланмаларни, услубий қўлланмаларни, тарқатма ва кўргазмали ашёларни аҳамияти каттадир.

### **Мустақил таълимни ташкил этишнинг шакли ва мазмуни**

«Дастурлаш асослари» фанини ўрганувчи талабалар аудиторияда олган назарий билимларини мустаҳкамлаш ва амалий масалаларни ечишда кўникма ҳосил қилиш учун мустақил таълим тизимига асосланиб, кафедра ўқитувчилари раҳбарлигида, мустақил иш бажарадилар. Бунда улар қўшимча адабиётларни ўрганиб ҳамда Интернет сайтларидан фойдаланиб рефератлар ва илмий докладлар тайёрлайдилар, амалий машғулоти мавзусига доир уй вазифаларини компьютерда бажарадилар, кўргазмали куроллар ва слайдлар тайёрлайдилар.

#### **2.1. ОБни баҳолаш**

Оралиқ назорат “Дастурлаш асослари” фанининг бир неча мавзуларини қамраб олган бўлими бўйича, тегишли назарий ва амалий машғулоти ўтиб бўлингандан сўнг ёзма равишда амалга оширилади. Бундан мақсад талабаларнинг тегишли саволларни билиши ёки муаммоларни ечиш кўникмалари ва малакалари аниқланади. Ўқув йилининг 1 ва 2-семестрларида 2тадан ОБ ўтказиш режалаштирилган бўлиб жами 5 балдан иборат. ОБ назорат ишлари ёзма иш усулида ўтказилиши назарда тутилган, ёзма иш соволлари ишчи ўқув дастур асосида тайёрланади. ОБ ни ўзлаштирмаган талабаларга қайта топшириш имконияти берилади. ОБ бўйича олинadиган ёзма ишлар кафедра мудири раҳбарлигида ташкил этилади ва кафедрада олти ой сақланади.

#### **2.4. ЯБни баҳолаш**

Якуний назорат “Дастурлаш асослари” фанининг барча мавзуларини қамраб олган бўлиб, назарий ва амалий машғулоти ўтиб бўлингандан сўнг ёзма равишда амалга оширилади. Бундан мақсад талабаларнинг фан бўйича ўзлаштириш кўрсаткичлари, яъни билим даражаси ёки муаммоларни ечиш кўникмалари ва малакалари аниқланади. ЯБ назорат ишлари тест усулида ҳам ўтказилиши назарда тутилган, тест соволлари ишчи ўқув дастури асосида тайёрланади. ЯБни ўзлаштирмаган талабаларга қайта топшириш имконияти берилади. ЯБ бўйича олинadиган ёзма иш вариантлари кафедра мудири раҳбарлигида тузилади ва деканатларга топширилади.

### **Фойдаланилаган дарслик ва ўқув қўлланмалар рўйхати Асосий адабиётлар**

1. Bjarne Stroustrup. The C++ Programming Language (3th Edition). Addison-Wesley, 1997.
2. D.S. Malik. C++ Programming: From Problem Analysis to Program Design. Fifth Edition. Course Technology, 2011.
3. Мадрахимов Ш.Ф., Гайназаров С.М. C++ тилида дастурлаш асослари// Тошкент, ЎзМУ, 2009, 196 бет.
4. Madraximov Sh.F., Ikramov A.M., Babajanov M.R. C++ tilida programmalash bo'yicha masalalar to'plami. O'quv qo'llanma // Toshkent, O'zbekist
5. on Milliy Universiteti, "Universitet" nashriyoti, 2014. - 160 bet.

### **Қўшимча адабиётлар**

1. Bjarne Stroustrup. The C++ Programming Language (4th Edition). Addison-Wesley, 2013. 1363 page.
2. Bjarne Stroustrup. Programming: Principles and Practice using C++ (Second Edition)" Addison-Wesley, 2014, 1305 page.
3. Павловская Т.А. C++. Программирование на языке высокого уровня – СПб.: Питер. 2005.- 461 с.
4. Walter Savitch. Absolute C++, 5th edition. Addison-Wesley/Pearson, 2012. 984 page.
5. Walter Savitch. Problem Solving with C++, 9th edition. Addison-Wesley/Pearson, 2015. 1088 page.
6. Павловская Т.С. Щупак Ю.С. C/C++. Структурное программирование. Практикум.-СПб.: Питер,2002-240с
7. Глушаков С.В., Коваль А.В., Смирнов С.В. Язык программирования C++: Учебный курс.- Харьков: Фолио; М.: ООО «Издательство АСТ», 2001.- 500с.
8. Культин Н.Б. C++Builder в задачах и примерах.-СПб.: БХВ-Петербург, 2005.-336с.
9. Абрамов С.А., Гнезделова Капустина Е.Н. и др. Задачи по программированию. - М.: Наука, 1988.

### **Электрон манбалар**

1. <http://cppstudio.com> – C++ тилида программалаш бўйича намуналар изоҳлари билан келтирилган
2. <http://cplusplus.com> – C++ тилида мавжуд конструкциялар таърифи, ишлатиш намуналари билан келтирилган.
3. <http://www.compteacher.ru/programming> – дастурлаш бўйича видео дарсликлар мавжуд.
4. <http://www.intuit.ru> – интернет университет, дастурлаш бўйича ёзма ва видео маърузалар ўқиш, тест синовларидан ўтиш ва сертификат олиш имконияти мавжуд.
5. <http://www.ziyonet.uz> – дастурлаш асослари бўйича рефератлар топиш мумкин.

\_\_\_\_\_ ўқув йили учун ишчи ўқув дастурига қўйидаги ўзгартириш ва қўшимчалар киритилмоқда:

[illegible]

(профессор-ўқитувчининг И.Ф.О.)

(ИМЗОСИ)

Ишчи ўқув дастурга киритилган ўзгартириш ва кўшимчалар “Физика-математика” факультети Илмий-услубий Кенгашида муҳокама этилди ва маъқулланди ( \_\_\_\_ йил “ \_\_\_\_ ” даги “ \_\_\_\_ ” - сонли баённома).

Факультет Илмий-услубий  
Кенгаши раиси:

доц.Ш.А.Аширов



## Тест топшириқлари

Сатрни тўғри эълон қилиш

```
Char a[10];
```

```
a[10] Char;
```

```
Char a;
```

```
a Char;
```

Сатрни тўғри эълон қилиш

```
Char b[10];
```

```
b[10] Char;
```

```
Char b;
```

```
b Char;
```

Сатрни тўғри эълон қилиш

```
Char a[10],b[100];
```

```
a[10] Char b[100];
```

```
Char a,b;
```

```
a Char b;
```

Сатрни реал узунлигини аниқлаш функцияси

```
Strlen
```

```
Strin
```

```
Sizeof
```

```
Str
```

Strlen функциясининг вазифаси

Сатрни реал узунлигини аниқлаш

Сатрни узунлигини эълон қилиш

Сатрни улаш

Сатрни эълон қилиш

Нол-терминаторни ҳисобга олган ҳолда сатр узунлигини аниқлаш  
функцияси

```
Sizeof
```

```
Strlen
```

```
Strin
```

```
Str
```

Sizeof функциясининг вазифаси

Нол-терминаторни ҳисобга олган ҳолда сатр узунлигини аниқлаш

Сатрни реал узунлигини аниқлаш

Сатрни узунлигини эълон қилиш

Сатрни улаш

```
Натижани аниқланг char a[]="0123456789"; cout<<strlen(a);
```

```
10
```

```
11
```

```
9
```

```
0123456789
```

```
Натижани аниқланг char a[]="Uzbekistan"; cout<<strlen(a);
```

```

10
11
9
Uzbekistan
Натижани аниқланг char a[]="Gulistan"; cout<<strlen(a);
8
7
9
Gulistan
Натижани аниқланг char a[]="0123456789"; cout<<sizeof(a);
11
10
9
0123456789
Натижани аниқланг char a[]="Uzbekistan"; cout<<sizeof(a);
11
10
9
Uzbekistan
Натижани аниқланг char a[]="Gulistan"; cout<<sizeof(a);
9
7
8
Gulistan
Сатрларни нусхалаш функцияси
Strcpy
Strncpyf
Str
cpy
Strcpy функциясининг вазифаси
Сатрни нусхалаш
Сатрни ўчириш
Сатр узунлигини аниқлаш
Сатрни излаш
Strncpy функциясининг вазифаси
Бир сатрдан иккинчисига бир нечта белгини нусхалаш
Сатрни ўчириш
Сатр узунлигини аниқлаш
Сатрни излаш
Қайси функция бир сатрдан иккинчисига бир нечта белгини нусхалайди
Strncpy
Strcpy
Strncpy
Strcp

```

Strcpy va Strncpy функцияларининг фарқи нимада  
 Strncpy бир сатрдан иккинчисига бир нечта белгини нусхалайди  
 Фарқи йўқ  
 Бундай операторлар йўқ  
 Strncpy функцияси сатрдан N та белгини ўчиради  
 Натижани аниқланг char a[]="123456", b[]="0123456789"; strncpy(a,b,3);  
 cout<<a;  
 012456  
 0123456  
 123456  
 654321  
 Натижани аниқланг char a[]="Guliston", b[]="Uzbekiston"; strncpy(a,b,2);  
 cout<<a;  
 Uzliston  
 Guliston  
 Gubekiston  
 Uzbekiston  
 Натижани аниқланг char a[]="Ali", b[]="Vali"; strncpy(b,a,2); cout<<b;  
 Alli  
 Valli  
 Vai  
 Ai  
 Натижани аниқланг char a[]="5+5=10", b[]="2\*25=50"; strncpy(a,b,2);  
 cout<<a;  
 2\*5=10  
 5\*25=125  
 5+25=30  
 5\*2=10  
 Натижани аниқланг char a[25]="Uzbekiston", b[]="0123456789"; strcat(a,"  
 vatanim"); cout<<a;  
 Uzbekiston vatanim  
 Uzbekiston vatani  
 Uzbekiston vatan  
 vatanim Uzbekiston  
 Натижани аниқланг char a[25]="Uzbekiston", b[]="0123456789"; strcat(a,"  
 vatanim"); cout<<b;  
 0123456789  
 Uzbekiston vatanim  
 vatanim Uzbekiston  
 123456789  
 Натижани аниқланг char a[25], b[]="0123456789"; strcpy(a,"12345");  
 cout<<a;  
 12345  
 01234

012345

1234

Натижани аниқланг char a[25], b[]="012345"; strcpy(a,"12345"); cout<<b;

012345

01234

12345

1234

Натижани аниқланг char a[25]="Рақамлар", b[]="0123456789";

strncat(a,b,5); cout<<a;

Рақамлар01234

Рақамлар12345

Рақамлар012345

Рақамлар1234

Натижани аниқланг char a[25]="Рақамлар", b[]="0123456789";

strncat(b,a,5); cout<<b;

0123456789Рақам

Рақам0123456789

0123456789Рақамлар

Рақамлар0123456789

Strcmp функциясининг вазифаси нима

Сатрларни солиштириш

Сатрларни нусхалаш

Сатрларни улаш

Сатрни ўчириш

Қайси функция сатрларни солиштириш имконини беради

Strcmp

Strcpy

Strcat

Strlen

Strcat функциясининг вазифаси нима

Сатрларни улаш

Сатрларни нусхалаш

Сатрни ўлчаш

Сатрни ўчириш

Қайси функция сатрларни улаш имконини беради

Strcat

Strcmp

Strcpy

Strlen

Strncat функциясининг вазифаси нима

Бир сатрга иккинчисидан бир нечта символни улаш

Сатрларни нусхалаш

Сатрга n та символни улаш

Сатрни ўчириш

```

Натижани аниқланг char a[]="012345"; int i, s=0; cout<<strlen(a);
6
5
012345
12345
Натижани аниқланг char a[]="012345"; int i, s=0; cout<<sizeof(a);
7
6
012345
5
Натижани аниқланг char a[]="012345"; int i, s=0; for (i=1;i<=strlen(a);i++)
s+=i; cout<<s;
21
15
10
18
Натижани аниқланг char a[]="012345"; int i, s=0; for (i=1;i<=sizeof(a);i++)
s+=i; cout<<s;
28
15
10
21
Натижани аниқланг char a[]="012345"; int i, s=0; for (i=1;i<strlen(a);i++)
s+=i; cout<<s;
15
21
10
18
Натижани аниқланг char a[]="012345"; int i, s=0; for (i=1;i<sizeof(a);i++)
s+=i; cout<<s;
21
15
10
18
Натижани аниқланг char a[]="Uzbekistan Tashkent"; int s=0,i; for
(i=1;i<strlen(a);i++) if (a[i]=='a' || a[i]=='k') s++; cout<<s;
4
5
0
3
C++ tilida konsol rejimda ishlash jarayonida berilganlarni standart oqimdan
o'qish formati to'g'ri ko'rsatilgan javobni aniqlang
cin>> <o'zgaruvchi>
cout<< <ifoda>

```

```
cin<< <o'zgaruvchi>
```

```
cout>><o'zgaruvchi>
```

C++ tilida konsol rejimda ishlash jarayonida ma'lumotlarni standart oqimga (ekranga)chiqarish uchun to'g'ri format keltirilgan javobni toping

```
cout<< <ifoda>
```

```
cin>> <o'zgaruvchi>
```

```
cout>> <ifoda>
```

To'g'ri javob keltirilmagan

C++ tilida konyunksiya(mantiqiy ko'paytma) amali qaysi belgi yordamida ifodalanadi

&&

||

!

%

C++ tilida mantiqiy inkor amali qaysi belgi yordamida ifodalanadi

!

||

&&

^

Hisoblashda bajariladigan amallar tartibi ko'rsatilsin?  $a / b + a \% b * c$

/, %, \*, +

%, /, \*, +

%, /, +, \*

/, +, %, \*

Quyidagi programma qismi nima natija chiqaradi? `int n=7; bool k; k=n%2; cout << k;`

1

true

false

0

Quyidagi programma qismi nima natija chiqaradi? `int n=123,a ; a=n%100; cout << a;`

23

12

3

0.3

Quyidagi programma qismi nima natija chiqaradi? `int a=17; cout << a%10<<a/10;`

71

17

18

16

Uchburchakning tomonlari a, b, c bo'lsa, uchburchak mavjudligini tekshiruvchi tengsizlik qaysi javobda berilgan?

`t = (a<b+c) && (b<a+c)&&(c<a+b)`

`t = (a<b+c) || (b<a+c)`

`t = (a<b+c) && (b>a+c)&&(c<a+b)`

`t = (a<b+c) || (b<a+c)|| (c<a+b)`

C++ tilida shart oldin tekshiriluvchi takrorlash operatorini ko'rsating.

`while`

`for ( )`

`while...do`

`if ( )`

Quyidagi programma ishlashi natijasida ekranda qanday qiymat aks etadi? `int`

`s=1; for(int i= 0;i<=20; i++) {s+=i; if(s == 10) break;} cout << s ;`

211

191

10

1

Postfiks amali to'g'ri yozilgan javobni aniqlang

`x=y++`

`x=++y`

`x+=y`

`x=x+y`

C++ tilde `while` takrorlash operatori qachongacha operator yoki blokni takror bajaradi?

Takrorlash sharti yolg'on(false yoki 0) bo'lguncha

Takrorlash sharti rost (true) bo'lguncha

Takrorlash sharti bir (1) bo'lguncha

O'zidan keyin yana takrorlash operatori uchramaguncha

C++ tilida `for` takrorlash operatorining sintaksis ko'rinishi to'g'ri ko'rsatilgan javobni toping.

`for(<ifoda1>;<ifoda2>; <ifoda3>)<operator yoki blok>`

`for(<ifoda>)operator yoki blok>`

`for(<ifoda1>,<ifoda2>, <ifoda3>)<operator>`

`for(<ifoda1>;<ifoda2>)<operator>`

`int b = 23; a = 30; a+ = a + b ++; a ning qiymatini hisoblang.`

83

54

55

80

`int b = 54; a = 30; a- = a + b --; a ning qiymatini hisoblang.`

-54

84

-50

80

Quyidagi programma ishlashi natijasida ekranda qanday qiymat aks etadi? `int`

`s=0,i; for(i=1;i<10;i++) s+=i; cout<<s;`

45

55

10

1

Quyidagi programma ishlashi natijasida ekranda qanday qiymat aks etadi? int

a=10, b=23; switch(a){ case 1: c=a+b; break; case 2: c = a\*b, break; default:

c = (a + b)\*b, break; } cout<<c;

759

33

230

xatolik haqida habar beradi

Quyidagi programma qismi nima natija chiqaradi? int a; bool t=true, T=true; if(t

&& T) a = 50; else a=200; cout <<a;

50

100

200

150

Quyidagi programma qismi nima natija chiqaradi? int a; bool t=true, T=false;

if(t && T) a=100; else a=200; cout <<a;

200

50

150

300

Quyidagi programma qismi nima natija chiqaradi? int a; bool t=false, T=false;

if(t && T) a=100; else a=200; cout <<a;

200

100

400

500

Quyidagi programma qismi nima natija chiqaradi? int a; bool t=false, T=true;

if(t || T) a=500; else a=800; cout <<a;

500

1000

800

1300

Quyidagi programma ishlashi natijasida ekranda qanday qiymat aks etadi? int

a=10, b; if(a%2==0) b = a\*a; else b=-a; cout<<b;

100

-10

10

0

Quyidagi programma ishlashi natijasida ekranga qanday qiymat chiqadi? int

x=1; x+=5; cout<<x;

6



5

0

Xatolik haqidagi habar

Quyidagi programma ishlashi natijasida ekranga qanday qiymat chiqadi? int

a=4, b=8, y; y=a>b?a:b; cout<<y;

8

5

12

0.5