

2-mavzu. Borland C++ Builderda tiplar

Asosiy savollar

- 1. Borland C++ Builder da butun sonlar**
- 2. Borland C++ Builder da haqiqiy sonlar**

Dastur ishlashi jarayonida xotiradan kamroq joy egallashi dasturning tez ishlashiga olib keladi.

Bu muammolar dasturdagi o'zgaruvchilar sonini kamaytirish, yoki o'zgaruvchilar saqlanadigan yacheyka hajmini kamaytirish orqali erishiladi.

Borland C++ Builder da butun va haqiqiy sonlarni e'lon qilish uchun bir nechta toifalar mavjud. Ular bir - biridan kompyuter xotirasida qancha hajm egallashi va qabul qiluvchi qiymatlar oralig'i bilan farq qiladi.

Butun sonlar

unsigned short int 0..65535 2 bayt

short int –32768..32767 2 bayt

unsigned long int 0..42949667295 4 bayt

long int –2147483648..2147483647 4 bayt

int (16 razryadli) –32768..32767 2 bayt

int (32 razryadli) –2147483648..2147483647 4 bayt

unsigned int (16 razryadli) 0..65535 2 bayt

unsigned int (32 razryadli) 0..42949667295 4 bayt

Haqiqiy sonlar

float 1.2E–38..3.4E38 4 bayt

double 2.2E–308..1.8E308 8 bayt

long double (32 razryadli) 3.4e-4932..-3.4e4932 10 bayt

Mantiqiy ifoda

bool true yoki **false** 1 bayt

Belgilar

char 0..255 1 bayt

void 2 yoki 4

Har xil toifadagi o'zgaruvchilar kompyuter xotirasida turli xajmdagi baytlarni egallaydi. Xattoki bir toifadagi o'zgaruvchilar ham qaysi kompyuterda va qaysi operatsion sistemada ishlashiga qarab turli o'lchamdagi xotirani egallashi mumkin.

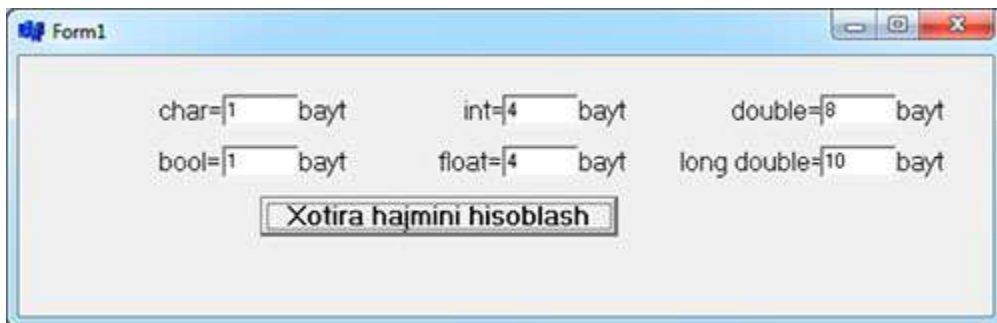
Borland C++ Builder da ixtiyoriy toifadagi o'zgaruvchilarning o'lchamini sizeof funksiyasi orqali aniqlash mumkin. Bu funksiyani o'zgarmasga, biror toifaga va o'zgaruvchiga qo'llash mumkin.

Toifalarni kompyuter xotirasida egallagan xajmini aniqlash.

Borland C++ Builder da dasturu:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    Edit1->Text=sizeof(char);
    Edit2->Text=sizeof(bool);
    Edit4->Text=sizeof(int);
    Edit5->Text=sizeof(float);
    Edit6->Text=sizeof(double);
    Edit7->Text=sizeof(long double);
}
```

NATIJA:



Matematik funksiyalardan dasturda foydalanish uchun math.h sarlavha faylini progarmmaga qo'shish kerak. #include <math.h>

1. abs(x) - butun sonlar uchun /x/
 2. fabs(x) - haqiqiy sonlar uchun
 3. labs(x) - uzun butun son uchun
 4. Pow(x,y)- x^y
 5. sqrt(X)-ildiz(X)
 6. ceil(x)-haqiqiy toifadagi x o'zgaruvchisi qiymatini unga eng yaqin katta butun songa aylantiradi.
 7. floor(x)-haqiqiy toifadagi x o'zgaruvchisi qiymatini unga eng yaqin kichik butun songa aylantiradi.
 8. cos(x)-x burchak kosinusini aniqlash. x radian o'lchovida.
 9. sin(x)-x burchak sinusini aniqlash. x radian o'lchovida.
 - 10 exp(x)- e^x
 11. log(x)-x sonining natural logarifmini qaytaradi.
 12. log10(x)-x sonining 10 asosli logarifmini qaytaradi.
- 1 - Misol: n va m natural sonlari berilgan. n sonini m soniga bo'lib, qoldiqni aniqlovchi dastur tuzilsin.

Borland C++ Builder da dasturu:

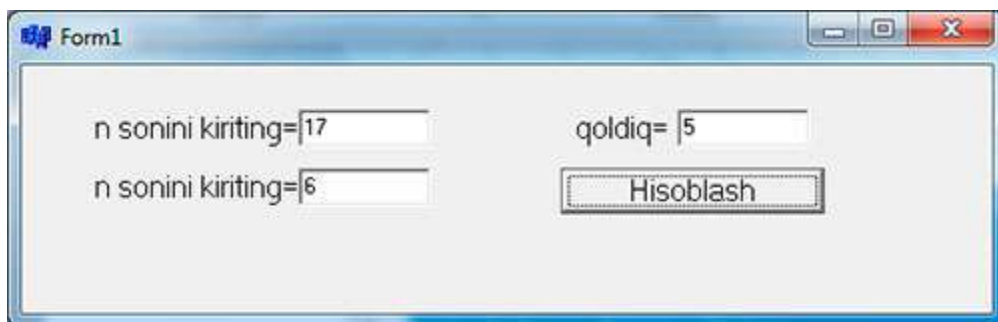
```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
```

```

int a,n,m;
n=StrToFloat(Edit1->Text);
m=StrToFloat(Edit2->Text);
a=n%m;
Edit3->Text=FloatToStr(a);
}

```

NATIJA:



Dasturlash san'ati

1. Har bir dastur, muallif, sana, dastur maqsadini anglatuvchi izoh bilan boshlanishi kerak.
 2. Dastur yozayotganda joy tashlashlarni kelishilgan, aniq bir qoida asosida olib borgan maqul. Masalan, tabulyatsiyani 4 ta probel deb qabul qilish mumkin. Ammo bu har kimning tasavvuriga bog'liq, maqsad shuki, dastur sodda oq'ishli va ko'rinishli bo'lsin.
 3. Har bir verguldan keyin probel tashlang, dastur oson o'qilsin.
 4. O' zgaruvchilarni e' lon qilishni boshqa operatorlardan bo' sh satr bilan ajratib qo'ying.
 5. (+, -, *, /) kabi amallarni har ikkala tomonidan probel qo'ying. Bu dastur o' qilishini qulaylashtiradi.
- 2 - Misol: n va m natural sonlari berilgan. n sonini m soniga bo'lib, butun qismini aniqlovchi dastur tuzilsin.

Borland C++ Builder da dasturu:

```

void __fastcall TForm1::Button1Click(TObject *Sender)
{
int a,n,m;
n=StrToFloat(Edit1->Text);
m=StrToFloat(Edit2->Text);
a=n/m;
Edit3->Text=FloatToStr(a);
}

```

NATIJA:

3 - misol. a sonini b soniga bo`lib 2 xona aniqlikda chiqarish.

Borland C++ Builder da dasturu:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
int a,n,m;
n=StrToFloat(Edit1->Text);
m=StrToFloat(Edit2->Text);
a=n/m;
Edit3->Text=FloatToStr(a);
}
```

NATIJA:

4 - misol. Butun sonni bo'lish

Borland C++ Builder da dasturu:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
int a,b;
a=StrToFloat(Edit1->Text);
b=StrToFloat(Edit2->Text);
Edit3->Text=FloatToStr((float(a))/(float(b)));
Edit4->Text=FloatToStr(a/b);
}
```

NATIJA:

The screenshot shows a Windows application window titled "Form1". Inside the window, there are four text input fields arranged in a 2x2 grid. The first row contains the labels "Bo'linuvchi sonni kiriting!" and "haqiqiy turdagi natija=" followed by input fields containing the values "1" and "0,5" respectively. The second row contains the labels "Bo'luvchi sonni kiriting!" and "butun turdagi natija=" followed by input fields containing the values "2" and "0" respectively. Below these fields, centered, is a button labeled "Hisoblash".

Bo'linuvchi sonni kiriting!	1	haqiqiy turdagi natija=	0,5
Bo'luvchi sonni kiriting!	2	butun turdagi natija=	0

Hisoblash