

12-mavzu: Tarmoqlanish operatorlari.

Reja:

1. IF tanlov operatori;
2. SWITCH tanlov operatori

C++ tilida ikkita tanlov operatori mavjud bo'lib ular quyidagilar:

- IF tanlov operatori;
- SWITCH tanlov operatori.

Tanlov operatori qo'yilgan shartning haqiqiy yoki yolg'on ekanligiga qarab qaror qabul qilish imkonini beradi. Agar qo'yilgan shart haqiqiy bo'lsa (**true**) u holda operator IF bajariladi, keyin esa undan keyingi operator amali bajariladi. Agar qo'yilgan shart yolg'on bo'lsa (**false**) u holda IF operator sharti bajarilmaydi yoki o'tkazib yuboriladi va darhol keyingi operator amali ishga tushadi.

Tekshirilayotgan shart har qanday mantiqiy ifoda bo'lishi mumkin. Mantiqiy amallar, tengsizlik yoki nisbatlar bo'lishi mumkin (quyidagi jadvalga qarang).

jadval. C++ tilida mantiqiy amallar

C++ tilida amallar	C++ tilida bajariladigan amallarga misollar	C++ tilida shartlarning mazmuni
==	a == b	a b ga teng
!=	a != b	a b ga teng emas
>	a > b	a b dan katta
<	a < b	a b dan kichik
>=	a >= b	a b dan katta yoki teng
C++ tilida amallar	C++ tilida bajariladigan amallarga misollar	C++ tilida shartlarning mazmuni
<=	a <= b	a b dan kichik yoki teng

Programmashda ko'pgina hollarda xatolik belgilarni noto'g'ri yozish tufayli paydo bo'ladi, masalan quyidagilar:

==; !=; >=; <=; belgilar orasida bo'shliq yozilmasligi zarur;

"!=" notog'ri; "!=" tog'ri;

">" notog'ri; ">=" tog'ri;

"<" notog'ri; "<=" tog'ri;

Bundan tashqari egalik qilish belgisi = bilan tenglik belgisini == almashtirish mumkin emas, chunki bunday xatolik mantiqiy xatolikka olib keladi. Bunday xatolik bilan programma kompilyatsiya qilinishi mumkin, ammo u notog'ri ishlaydi. Tanlov operatori **IF** ikki ko'rinishda yozilishi mumkin, ular quyidagilar:

```
if(/*tekshiriladigan shart*/)
{
```

//shart operatori yozilish joyi

}

Agar haqiqiy shartni bajarish bitta operatorda bajarilishi zarur bo'lsa u holda katta qavslarni tashlab yuborish mumkin, masalan:

if(/*tekshiriladigan shart*/)

/*1-operator*/;

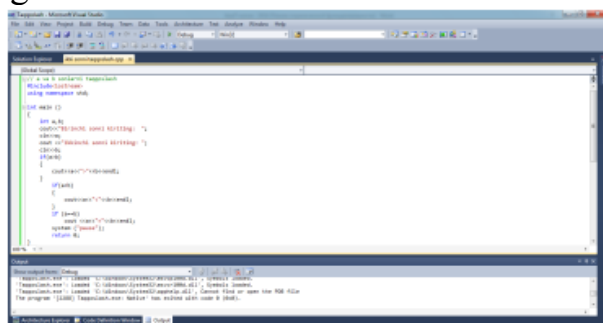
Quyida ikkita sonli taqqoslash programmasini ishlab chiqamiz va tahlil qilamiz.

Misolda ikkita **a** va **b** sonlari berilgan bo'lib ularni o'zaro taqqoslash zarur.

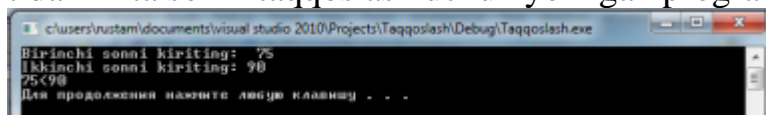
Quyidagi programmaga e'tibor beramiz (2.32 va 2.33 - rasmlar).

```
1 // in_out.cpp: konsol ilovaga kirish nuqtasi.
2 #include "stdafx.h"
3
4 #include <iostream>
5 using namespace std;
6 int main()
7 {
8
9     int a, b; // int turidagi o'zgaruvchini e'lon qilish, bunday turdagi
10    o'zgaruvchida butun sonlardan iborat ma'lumot saqlanadi
11    cout<<"Birinchi sonni kiriting: ";
12    cin>> a;
13    cout<<"Ikkinchi sonni kiriting: ";
14    cin>> b;
15    if(a > b) // agar a > b
16    {
17        cout<< a <<" > " << b <<endl; // bu xabarni bosmaga chiqaring
18    }
19    if(a < b)// agar a < b
20    {
21        cout<< a <<" < " << b <<endl;
22    }
23    if(a == b) // agar a = b
24    cout<< a <<" = " << b <<endl; // shartli operator tanasida katta qavs
25    yozilmagan
26    system("pause");
27    return 0;
28 }
```

1 dan 8 – qatorgacha sizga tushunarli bo'lishi zarur, chunki siz bu amallar bilan bundan oldingi mavzularda tanishdingiz. 9 – qatorda ikkita **a** va **b** o'zgaruvchilar e'lon qilingan. Programma sizdan birinchi va ikkinchi sonlarni kiritishingizni so'raydi keyin esa kiritilgan raqamlar taqqoslanadi. Programma foydalanuvchilarning qanday raqamlarni kiritishini bilmaydi, shuning uchun quyidagicha holatlar bo'lishi mumkin.



tida ikkita sonni taqqoslash uchun yozilgan programma



ikkita sonni taqqoslash uchun yozilgan programma natijasi

1-hol: birinchi son ikkinchi sondan katta bo'lishi mumkin (**14 – 16 -qatorlar**).

2-hol: birinchi son ikkinchi sondan kichik (**17 – 20 - qatorlar**).

3-hol: birinchi son ikkinchi songa teng (**21 – 23 - qatorlar**).

4-hol Programmaning 14 – 19 – qatorlariga e'tibor bering. Bu qatorlarda birinchi va ikkinchi shartli **IF** operatorlari tanasi katta qavs ichida yozilgan, ammo qavslarsiz ham yozish mumkin, chunki shart tog'ri bo'lgan holda faqat chiqish operatori bajariladi. Ko'p hollarda umumiy xatolik sifatida shuni ta'kidlash zarurki **IF** operatori qavsi yopilishidan keyin nuqta – vergul (;) qo'yiladi, bu esa sintaksis xatolik hisoblanadi. Quyidagiga e'tibor bering.

```
if(a == b); // mantiqiy xatolik
```

```
cout<< a <<" = " << b <<endl;
```

Bunday holda programma kompilyatsiya bo'lishi mumkin, ammo xato ishlaydi. Xatolik shundan iboratki unda ishlatilayotgan **IF** da uning tanasiga kirayotgan operator yo'q, yoki **IF** bo'sh holda qolmoqda. Shart haqiqiy yoki yolg'onligidan qat'iy nazar ma'lumotni chiqarish operatori **cout<<a<<" = "<<b<<endl;** ishlaydi.

Ko'pincha dasturni yozish jarayonida uzoq if-else konstruksiyalarini yozish talab qilinadi, masalan, foydalanuvchidan qandaydir kalit olganimizda; agar siz o'yin yozayotgan bo'lsangiz, o'yinchi qaysi tugmani bosganini tekshirishingiz kerak bo'ladi (o'ng, chap, bo'sh joy va boshqalar).

Switch-case konstruksiyasi o'zgaruvchini int yoki char kabi bir nechta doimiy qiymatlar bilan taqqoslaydigan uzoq if-else konstruksiyasini qulay almashtirishdir.

```
switch ( <o'zgaruvchi> ) {
```

```
case ifoda1:
```

```
    Agar <o'zgaruvchi> == igoda1 bajar
```

```
    break;
```

```
case ifoda2:
```

```
    Agar <o'zgaruvchi> == ifoda2 bajar
```

```
    break;
```

```
...
```

```
default:
```

```
    agar birorta ham variant to'g'ri kelmasa bajar
```

```
    break;
```

```
}
```

Qavslar ichidagi o'zgaruvchi case kalit so'zidan keyin e'lon qilingan qiymatlar bilan taqqoslanadi. Ikki nuqtadan keyin, agar o'zgaruvchi joriy qiymatga teng bo'lsa, bajariladigan kod. kommutatorning bajarilishini buzish uchun tanaffus kerak.

break: ishlatilmagan masalani ko'rib chiqamiz

```
int a=1;
```

```
switch(a)
```

```
{
```

```
    case 1:
```

```
        a++;
```

```
    case 2:
```

```

        a++;
    case 3:
        a++;
    }
    cout<<"a= "<<a;
    natija a=4 bo'ladi.

```

Kiritilgan songa mos ravishda hafta kunini chiqaruvchi dastur tuzish.

```

#include <iostream>
using namespace std;

int main()
{
    int n;
    cout << "Hafta kunini kiriting"<<endl;
    cin >> n;
    switch (n)
    {
        case 1: cout << "Dushanba"; break;
        case 2: cout << "Seshanba"; break;
        case 3: cout << "Chorshanba"; break;
        case 4: cout << "Payshanba"; break;
        case 5: cout << "Juma"; break;
        case 6: cout << "Shanba"; break;
        case 7: cout << "Yakshanba"; break;
        default: cout << "Bunday hafta kuni yoq";
    }
    cout << endl;

    return 0;
}

```

Tanlash operatorida bir necha qiymatga bir xil operator ishlatilishini ifodalash.

```

#include <iostream>
using namespace std;
int n;
cout << "1...10 oralig'idagi son kiriting:"<< endl;
cin >> n;
switch (n)
{
    case 1:
    case 3:
    case 5:
    case 7:

```

```
case 9: cout << "Toq son kiritildi"; break;
case 2:
case 4:
case 6:
case 8:
case 10: cout << "Juft son kiritildi"; break;
default: cout << "1 dan kichik yoki 10 dan katta son kiritildi";
}
return 0;
}
```