

6-Ma'ruza. Tarmoqlanish operatorlari: sonlar va belgilarni solishtirish

Ma'ruza rejasi:

- 6.1 O'tish operatori.
- 6.2 Shartli operatorning qisqa ko'rinishi.
- 6.3 Shartli operatorning uzun ko'rinishi.
- 6.4 Tanlash operatori.

Kalit so'zlar: *mantiqiy qo'shish va ko'paytirish, inkor amali, tarmoqlanish, shartli operator, o'tish operatori, tanlash operatori*

Ko'p masalalarning yechimi ma'lum bir shart yoki shartlarning qo'yilishiga qarab bajariladi. Bunday jarayonlarni tarmoqlanuvchi hisoblash jarayoni deyiladi. Tarmoqlanuvchi hisoblash jarayonlari tarkibida yana tarmoqlanish bo'lishi mumkin. Bundaylarni murakkab tarmoqlanuvchi jarayonlar deb ataladi. Algoritmik tilda kattaliklarning istalgan xossasi shu ondagi qiymatlari uchun bajarilishi yoki bajarilmasligi **shart** deyiladi. Masalan: $a = v$ tenglik uchun $a=3$ va $v=3,1$ bo'lganda shart bajarilmaydi. N tub son deyilsa va $N=19$ bo'lsa, shart bajariladi, $N=15$ bo'lganda shart bajarilmaydi.

Tarmoqlanuvchi jarayonlarni tashkil etishda shartsiz o'tish va shartli o'tish operatorlaridan foydalaniladi. Shunday jarayonlar mavjudki, shartning bajarilishiga qarab, dasturning u yoki bu qismiga o'tishga to'g'ri keladi. Bunday hollarda shartsiz o'tish operatori ishlatiladi.

1. Shartsiz o'tish operatori: **goto n;** bu yerda n – metka, belgi bo'lib, jarayon o'tishi kerak bo'lgan joyni ko'rsatadi. Metka xarf, son va ular aralashmasidan iborat bo'lishi mumkin. 1ta operatorga bir nechta metkalarni qo'yish mumkin. (Usta dasturchilar goto n operatoridan kamroq foydalanadilar.)

2. Shartli o'tish operatori: **if (shart) operator;** Uning ishlashi quyidagicha: agar shart rost bo'lsa keltirilgan operator bajariladi, agar shart yolg'on bo'lsa, keyingi qatorga o'tiladi. Ko'pincha bu ko'rinish ishlatilganda 2ta operatorlar aralashib ketmasligi uchun goto operatori ishlatildi. Agar if so'zidan keyin bir nechta operatorlar keladigan bo'lsa, ularni alohida {} qavslarga olinadi. (bu usul kamroq ishlatiladi) Masalan:

$$Y \equiv \begin{cases} \sin x, & \text{agar } x < 5 \\ \sqrt[3]{x^2}, & \text{agar } x \geq 5 \end{cases}$$

```

#include <iostream.h>
#include <math.h>
void main ( )
{ float x, y;           // x va u ning toifasi xaqiqiy
  cin >> x;             // x ning son qiymati kiritiladi
  if (x<5)               // agar x<5 bo'lsa
  { y=sin(x); goto cc; } // 1-funksiya ishlaydi
  y=pow(x, 2/3.);        // aks xolda 2-funksiya ishlaydi
cc: cout << "y=" << y << endl; // u ning javobi beriladi. ss-metka
    }                   // main funksiyasi berkitildi.

```

3. if (shart) 1-operator(lar); else 2-operator(lar);

Masalan: yukoridagi misolni kurib utamiz:

```

#include <iostream.h>           cin >> x;
#include <math.h>               if (x<5) y=sin(x); else y=pow(x,
void main ( )                  2/3.);
{ float x, y;                  cout << "y=" << y << endl; }
Izox: if – else konstruksiyasi ichida yana if – else konstruksiyasi ishlatilishi mumkin.

```

Bunda bir nechta if operatoridan iborat ichma-ich joylashgan konstruksiya xosil buladi. Bunday xollarda else sO'zi O'ziga yaqin turgan if ga tegishli buladi.

shart ? 1-operator (lar) : 2-operator (lar);

Masalan: $x < 5$? $y = \sin(x)$: $y = \text{pow}(x, 2/3.)$;

(agar shart rost bulsa, 1-operator, aks xolda 2-operator bajariladi)

Tanlash operatori – O'zgaruvchining kiymatiga karab kup tarmok ichidan bittasi tanlanadi. Bu operatorning kurinishi kuyidagicha:

```

switch (ifoda yoki o'zgaruvchi)
{ case 1-qiymat : operator(lar); break;
  case 2-qiymat : operator(lar); break;
  case n - qiymat : operator(lar); break;
  default : aks holdagi operator (lar); }

```

Masalan:

$$Y = \begin{cases} \sin x, & \text{agar } x=1 \\ \cos x, & \text{agar } x=2 \\ \text{tg } x, & \text{agar } x=3 \\ \sqrt{x}, & \text{boshka barcha xollarda } (x>0) \end{cases}$$

```
# include <iostream.h>
# include <math.h>
void main ( )
{ int x; float y;
cin >> x;
switch (x)
{ case 1 : y=sin(x); break;
case 2 : y=cos(x); break;
case 3 : tan(x); break;
default : y=sqrt(x); }
cout << "y="<<y<<"x="<<x<<endl;
getch ( ); }
```

Izox: 1) switch operatoridagi ifoda yoki o'zgaruvchi butun toifali bo'lishi shart!

2) switch operatori satrlaridagi break so'zi tushib kolsa, joriy case operatoridan keyingi case bloklari ichidagi ifodalar xam bajarilaveradi.

Tekshirilayotgan shartlar bir nechta bo'lishi xam mumkin. Bunday xollarda ularni murakkab shart deyiladi. Bunday shartlarni quyidagi mantiqiy amallar orkali ifoda etiladi:

&& - mantiqiy kupaytirish (va)

|| – mantiqiy kushish (yoki)

! - mantiqiy inkor (emas)

Masalan: $6 \leq x \leq 10$ bulsa, $(x \geq 6) \&\& (x \leq 10)$

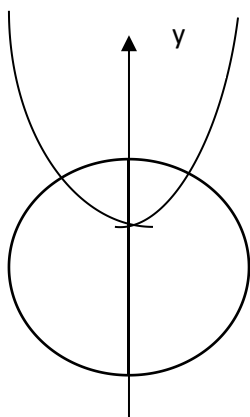
$Y > 0$ va $x < 4$ yoki $z \geq 5$ bylsa, $(y > 0) \&\& (x < 4) || (z \geq 5)$

if $(\sin(x) > 1) \&\& (5 / 2 = 0)$ y:=1; else y:=0; {ikkala shart xam yolg'on kiymatga ega, shuning uchun y=0 buladi.}

Mantiqiy kiymatlar ustida amallar bajarilganida quyidagi natijalar olinadi: (+ true 1; - false 0 degan ma'noda)

A	B	! A	!B	A && B	A B
+	+	-	-	+	+
+	-	-	+	-	+
-	+	+	-	-	+
-	-	+	+	-	-

Mantiqiy masala: ixtiyoriy berilgan $M(x,u)$ nukta $u=x^2$ va $x^2+u^2=4$ aylana bilan kesishgan sohaga yoki shu aylananing 4-choragi tashqarisiga tushishini tekshiring.



Демак:

$y > x^2$ and $x^2 + y^2 \leq 4$
or $x > 0$ and $y < 0$
and

$x^2 + y^2 \geq 4$

$x = 1, y = 1 \rightarrow \text{false}$

$x=1, y=0 \rightarrow \text{true}$

$x = -2, y = 0.5 \rightarrow \text{false}$

x

```

#include <iostream. h>
void main ( )
{ float x, y; int n;
cout << "nuqtaning koordinatalarini
kirting:";
cin >> x>>y;
if ((y>=x*x && x*x+y*y<=4) ||
3-misol:  $ax^2+bx+c=0$  kurinishdagi kvadrat tenglamaning hakikiy yechimlarini topish
algorithmi tO'zing. (  $a \neq 0$ ;  $b \neq 0$ ;  $c \neq 0$ ;)

```

