

7 - laboratoriya ishi

Mavzu: Dasturlarning kompyuter tomonidan bajarilishi. Translyatorlar va interpretatorlar. Takrorlanuvchi algoritmlarga dastur tuzish. *for*, *while*, *do while* operatorlaridan foydalanib dastur tuzish.

Ko'pincha masalalarni dasturlashda bir xil buyruqlar ketma-ketligini bir necha bor bajarilishga to'g'ri keladi. Bunday jarayon sikl deb ataladi. Muayyan buyruqlar ketma-ketligi yangi qiymatlar bilan bir necha marta takrorlansa, bunday algoritmlar takrorlanuvchi(siklik) algoritm deyiladi. C++ tilida takrorlanuvchi algoritmlarni dasturlash uchun maxsus sikl operatorlari mavjud: *for*, *while*, *do while*. Takrorlanuvchi algoritmlarni dasturlash uchun, shuningdek, shartli o'tish operatoridan ham foydalanish mumkin.

1. **For operatori** takrorlanishlar soni ma'lum bo'lganda ishlatiladi. For operatoriining umumiy yozilishi quyidagicha:

```
for (initsializatsiya; ifoda; o'sish)
{
    // operatorlar ketma-ketligi
    // ...
}
bu yerda
```

initsializatsiya- o'zlashtirish operatori bo'lib, unda sikl o'zgaruvchisining boshlang'ich qiymati beriladi. ushbu o'zgaruvchi sikl ishini bosharuvchi xisoblagich vazifasini bajaradi;

ifoda - sikl o'zgaruvchisining qiymati tekshiriladigan shartli ifoda. Ushbu bosqichda siklning keyingi bajarilishi aniqlanadi;

o'sish - har bir iteratsiyadan keyin sikl o'zgaruvchisining qiymati qanday o'zgarishini aniqlaydi.

for operatori *ifoda rost* qiymatni qabul qilsa, bajarilaveradi. *Ifodaning* qiymati *yolg'on* bo'lgandan so'ng to'xtaydi va *for* dan keyingi operator bajariladi.

1-misol. 100 dan 300 gacha bo'lgan butun sonlarning yig'indisini toping. Ushbu masalaning dasturi quyidagicha bo'lagi.

```
#include <iostream>
using namespace std;
int main()
{
    int sum,i;
    sum = 0;
```

```
for(i = 100; i <= 300; i++)
sum = sum + i;
cout<<"summa="<<sum;
return 0; }
```

2-misol. n natural son berilgan. Quyidagi yig`indini xisoblang:

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

Masalaning dasturi:

```
# include <iostream>
using namespace std;
int main()
{ // s = 1 + 1/2 + 1/3 + ... + 1/n
int n;
cout<<"n ni kiriting="; cin>>n;
float s = 0;
for (int i = 1; i<=n; i++)
s = s + 1.0/i;
cout<<"\n s="<<s;
return 0; }
```

while operatori.

Sikldagi takrorlanishlar soni noma'lum bo'lib, ma'lum shartga bog'liq bo'lsa, siklni tashkil qilishda **while** operatoridan foydalanish mumkin.

while operatorining umumiy shakli quyidagicha:

```
while (ifoda)
{
    // operatorlar ketma-ketligi
    // ...
}
```

bu erda, **ifoda**- C ++ tilidagi mantiqiy ifodadir. Operatorlar ketma-ketligi **ifoda rost** qiymatni qabul qilsa bajariladi. **Ifoda yolg'on** qiymatni qabul qilgandan so'ng, **while** operatorining bajarilishi to'xtaydi va boshqaruv **while** dan keyingi operatorga o'tadi.

3-misol. Haqiqiy a soni berilgan. Quyidagi shartni qanoatlantiruvchi eng kichik n ni toping

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} > a$$

Dasdtlab, yig'indining qiymati a dan kichik. Har bir iteratsiyada yig'indining qiymati oshib boradi. Ma'lum bir qadamdan keyin bu miqdor a dan katta bo'ladi. n ning shu shart bajariladigan eng kichik qiymatini topish kerak. Masalani yechish dasturi quyidagicha:

```
#include <iostream>
using namespace std;
int main()
{
    float a,sum;
    int n;
    // a va n ni kiritish
    a = 2.2;
    n = 1;
    sum = 1.0/n;
    while (sum < a)
    {
        n++;
        sum = sum + 1.0/n;
    }
    cout<<"\n sum="<<sum;
    return 0; }
```

4-misol. Natural son berilgan. Undagi 3 raqamlari sonini aniqlang.

Masalani yechish dasturi:

```
#include <iostream>
using namespace std;
int main()
{
    int n; // berilgan natural son
    int k; // berilgan sondagi 3 raqamlari soni
    int t, d; // qo'shimcha o'zgaruvchilar
    // n ni kiritish
    cout<<"n sonini kiriting"; cin>>n;
    t = n; // n ning nusxasini olish
    k = 0;
    while (t > 0)
```

```

{
    d = t % 10; // oxirgi raqamni ajratib ko'rsatish
    if (d == 3) k ++;
    t = t / 10; //oxirgi raqamni o`chirish
}
cout<<n<<" soniagi 3 raqamlar= "<<k<<" ta";
return 0; }

```

Do... while operatori.

Bu operator takrorlanishlar soni ma'lum bo'lmagan sikllarni tashkil etishda ishlatiladi. Bu operatorning ishlash jarayonida har bir qadamdan keyin shart tekshiriladi. **do ... while** operatori da takrorlanishlar kamida bir marta bajariladi. **do... while** opratorida **for** va **while** operatorlaridan farqli ravishda shart sikldan chiqishda tekshiriladi.

do ... while operatorining umumiy shakli:

```

do
{
    // operatorlar ketma-ketligi
    // ...
}
while (ifoda);

```

bu erda, **ifoda** - sikl o'zgaruvchisining qiymati tekshiriladigan ifoda. Ushbu bosqichda siklning keyingi bajarilishi aniqlanadi. Katta qavslarni qo'yish shart emas.

Sikl quyidagicha ishlaydi. Dastlab, sikl tanasi bajariladi. Keyin ifoda qiymati (shartli ifoda) tekshiriladi. Agar ifoda qiymati rost (true) bo'lsa, sikl tanasi yana bajariladi. Ifodaning qiymati yolg'on (false) bo'lgandan so'ng, sikl to'xtaydi.

5-misol. Argument x a dan b gacha h qadam bilan o'zgarganda $Y = e^{\cos x} + \log_c(x + 4)$ funksiya qiymatlari hisoblansin. Dasturni **for**, **while** va **repeat** oparatorlaridan foydalanib quyidagi uch xil ko'rinishda yozish mumkin:

```

#include <iostream>
#include <math.h>
using namespace std;
int main()
{

```

```

float a,b,h,c,y,x;
cout<<"a,b,h,c larni kiriting \n"; cin>>a>>b>>h>>c;
x=a;
do
{
y=exp(cos(x))+log(x+4)/log(c);
cout<<" x="<<x<<"  y="<<y<<endl;
x+=h;
}
while (x<=b);
return 0; }

```