

## 8,9-Ma'ruza. Takrorlanuvchi jarayonlar va ularni dasturlash. For, while va do operatorlari yordamida takrorlanishlarni dasturlash

### Ma'ruza rejasi:

#### 8.1 Sikl operatorlari

#### 8.2 O'tish operatorlari

Kalit so'zlar: *delete, masofa keltirish, delete[], new, indeks, this, indeksirlash, [] bo'sh xotira, void\*, konteyner, ro'yxat, manzil, nolinchi ko'rchsatkich, tugun, adres olish &, bo'shatish, ko'rsatkich, virtual destruktor, xotira, xotira chiqishi, destruktor, toifani o'zlashtirish, resurslar chiqishi, a'zo destruktori.*

### 8.1 Sikl operatorlari

**Oldingi shartli while operatori.** Oldingi shartli while operatori quyidagi umumiy ko'rinishga egadir:

while(ifoda)

Operator

Bu operator bajarilganda avval ifoda hisoblanadi. Agar uning qiymati 0 dan farqli bo'lsa operator bajariladi va ifoda qayta hisoblanadi. To ifoda qiymati 0 bo'lmaguncha sikl qaytariladi.

Agar dasturda while (1); satr qo'yilsa bu dastur hech qachon tugamaydi.

Misol. Berilgan n gacha sonlar yig'indisi.

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
long n,i = 1,s = 0;
```

```
scanf("%d",&n);
```

```
while (i<= n )
```

```
    s+= i++;
```

```
printf("\n s = %d",s);
```

```
}
```

Bu dasturda  $s+= i++$  ifoda  $s = s+i$ ;  $i = i+1$  ifodalarga ekvivalentdir.

Quyidagi dastur to nuqta bosilmaguncha kiritilgan simvollar va qatorlar soni hisoblanadi:

```
#include <stdio.h>                                ++nc;
int main()                                          if (c == '\n') ++nl;
{                                                  };
int nc = 0,nl = 0;                                printf("satrlar = %d simvollar = %d
char c;                                           \n",nl,nc);
while ((c = getchar()) != ' ')                    return 0;
{                                                  }
```

**Keyingi shartli do-while operatori.** Keyingi shartli do-while operatori umumiy ko'rinishi quyidagicha:

```
do
Operator
while(ifoda)
```

Sikl operatorining bu ko'rinishida avval operator bajariladi so'ngra ifoda hisoblanadi. Agar uning qiymati 0 dan farqli bo'lsa operator yana bajariladi va hokazo. To ifoda qiymati 0 bo'lmaguncha sikl qaytariladi.

Misol. Berilgan n gacha sonlar yig'indisi.

```
#include <stdio.h>                                s+= i++;
int main()                                          while (i<= n);
{                                                  printf("\n s = %d",s);
long n,i = 1,s = 0;                                return 0;
scanf("%d",&n);                                    }
```

do  
Bu dasturning kamchiligi shundan iboratki agar n qiymati 0 ga teng yoki manfiy bo'lsa ham, sikl tanasi bir marta bajariladi va s qiymati birga teng bo'ladi.

Parametrli for operatori. Parametrli for operatori umumiy ko'rinishi quyidagicha:

```
for( 1-ifoda;2- ifoda; 3-ifoda)
```

Operator

Bu operator quyidagi operatorga mosdir.

```
1-ifoda;
```

```
while(2-ifoda) {
```

```
operator
3-ifoda
}
```

Misol. Berilgan n gacha sonlar yig'indisi.

```
#include <stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int s = 0;
    for(int i = 1; i <= n; i++) s += i;
    printf("\n%d",s);
    return 0;
}
```

**Siklda bir nechta schyotchikni qo'llanilishi.** Parametrli for siklining sintaksisi unda bir nechta o'zgaruvchi - schyotchikni qo'llanilishiga, siklni davom etishini murakkab shartlarini tekshirishga va sikl schyotchiklari ustida ketma-ket bir nechta operatsiyani bajarilishiga imkon beradi.

Agarda bir nechta schyotchikka qiymat o'zlashtirilsa yoki ular o'rtasida bir nechta operatsiya bajarilsa, bu ifodalar vergul bilan ajratilgan holda ketma – ket yoziladi.

for siklida bir nechta schyotchikni qo'llanilishi

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int i,j;
    for (i = 0, j = 0; i < 3; i++, j++)
        printf("i:%d j:%d\n",i,j);
    getch();
    return 0;
}
```

Hatija:

```
i: 0      j: 0
i: 1      j: 1
i: 2      j: 2
```

## 8.2 O'tish operatorlari

**Uzish break operatori.** Ba'zi hollarda sikl bajarilishini ixtiyoriy joyda to'xtatishga to'g'ri keladi. Bu vazifani break operatori bajarishga imkon beradi. Bu operator darhol sikl bajarilishini to'xtatadi va boshqaruvni sikldan keyingi operatorlarga uzatadi.

Misol:

```
#include <stdio.h>
int main()
{
    int n;
```

```

while(1)
{
scanf("%d",&n);
if(n == 1||n == 0) break;
}
printf("Sikl tugadi");
return 0;

```

Bu misolda while(1) operatori yordamida cheksiz sikl hosil qilinadi. Agar 1 yoki 0 soni kiritilsa sikl to'xtatiladi.

**Qaytarish continue operatori.** Sikl bajarilishiga ta'sir o'tkazishga imkon beradigan yana bir operator continue operatoridir. Bu operator sikl qadamini bajarilishini to'xtatib for va while da ko'rsatilgan shartli tekshirishga o'tkazadi.

Misol:

```

#include <stdio.h>
int main()
{
int n;
for(;;)
{
scanf("%d",&n);
if(n == 1||n == 0) continue;
break;
printf("Sikl tugadi");
return 0;
}
}

```

Bu misolda for(;;) operatori yordamida cheksiz sikl hosil qilinadi. Agar 1 yoki 0 sonlardan farqli son kiritilsa sikl to'xtatiladi.

**O'tish operatori goto.** O'tish operatorining ko'rinishi:

goto <identifikator>. Bu operator identifikator bilan belgilangan operatorga o'tish kerakligini ko'rsatadi.

Misol uchun goto A1;...;A1:y = 5;

Strukturali dasturlashda goto operatoridan foydalanmaslik maslahat beriladi. Lekin ba'zi hollarda o'tish operatoridan foydalanish dasturlashni osonlashtiradi.

Misol uchun bir necha sikldan birdan chiqish kerak bo'lib qolganda, to'g'ridan-to'g'ri break operatorini qo'llab bo'lmaydi, chunki u faqat eng ichki sikldan chiqishga imkon beradi.

```

#include <stdio.h>
int main() {int n = 16,s = 0;
int i,j; for(i = 1;i<5;i++)
for(j = 1;j<5;j++) {
if(i*j>n) goto A;
C++; } A:printf("Sikl tugadi s
= %d",s);return 0;
}

```