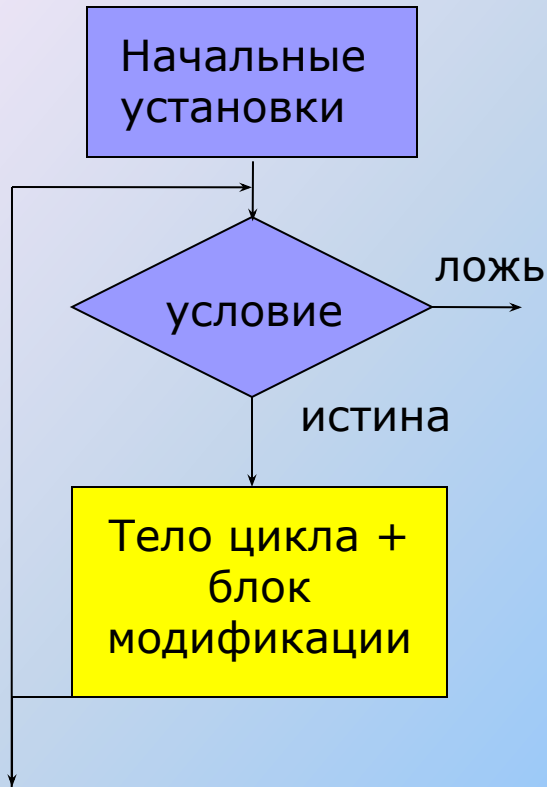


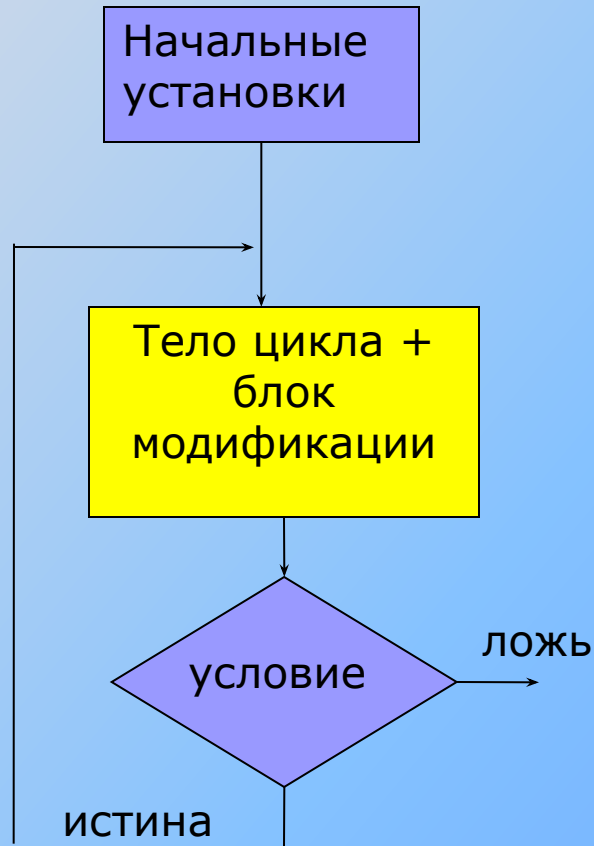
Модуль 4. Операторы цикла.

**Рассматриваются основные операторы
цикла**

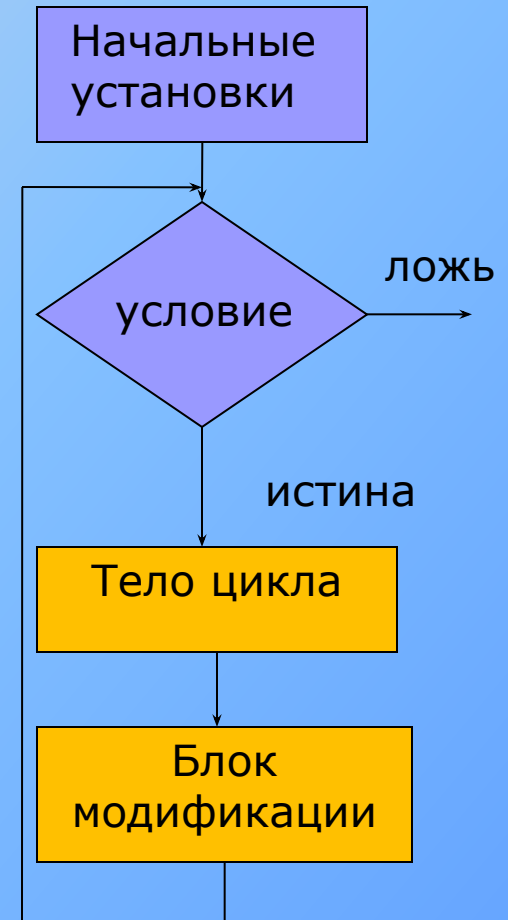
Циклические конструкции



А. Цикл с предусловием



Б. Цикл с постусловием

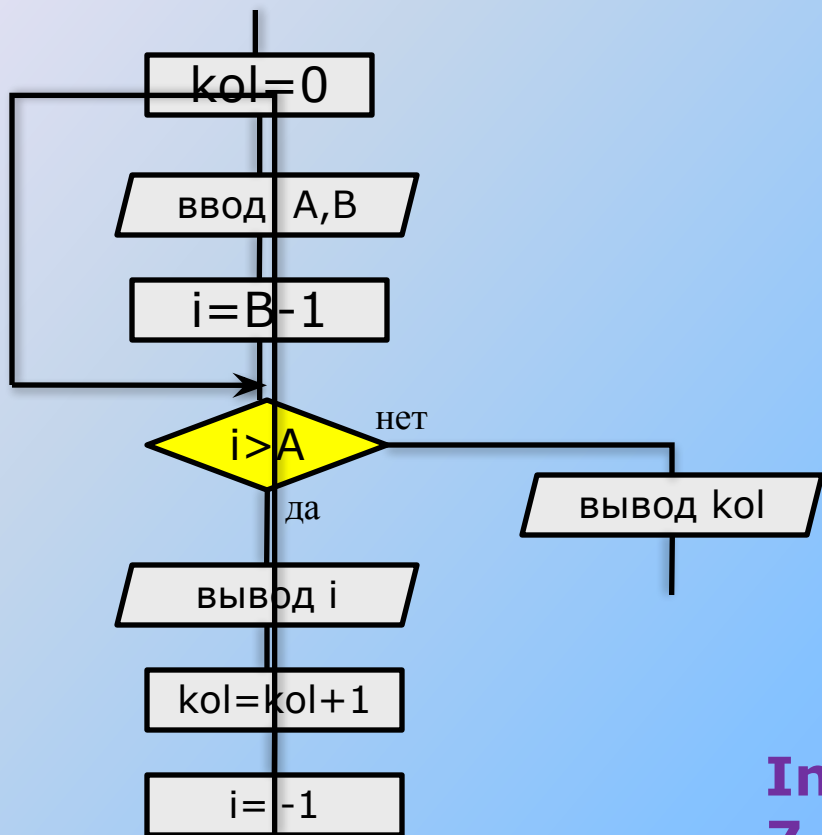


В. Цикл с параметром (со счетчиком)

Цикл с предусловием. Оператор WHILE. Пример 1.

while (выражение) оператор

Даны 2 целых числа A и B ($A < B$). Вывести в порядке убывания все целые числа, расположенные между A и B, и количество таких чисел.



```
int a,b,kol=0,i;
printf("\n Input a,b: ");
scanf("%d%d",&a,&b);
i=b-1;
while (i>a)
{
    printf("\n %d",i);
    kol++;
    i--;
}
printf("\nNumber of natural %d",
kol);
```

Input a,b: 5 8

7

6

Number of natural 2

Оператор WHILE. Пример 2.

Вывести таблицу значений функции $Y=X^2 + 1$ на отрезке $[x1;x2]$ с шагом dx .

```
#include <stdio.h>
int main() {
    float x1, x2, dx;
    printf("Введите диапазон и шаг изм-я аргумента: ");
    scanf("%f%f%f", &x1, &x2, &dx);
    printf("|   X   |   Y   |\n");
    float x = x1;
    while (x <= x2) {
        printf("| %5.2f | %5.2f |\n", x, x*x + 1);      x
        x += dx;
    }
}
```

```
while (int x = 0) { /* область действия x */ }
```

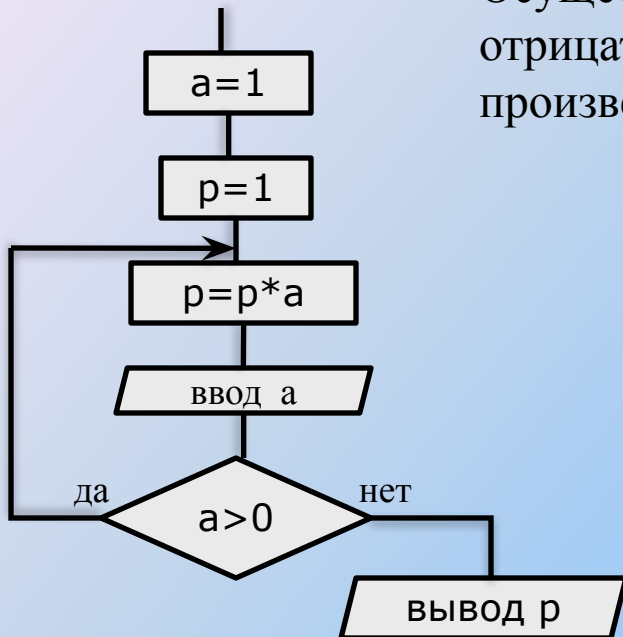
Цикл с постусловием. Оператор do while. Пример 1.

do оператор **while** выражение;

```
#include <iostream.h>
int main(){
    char answer;
    do{
        cout << "\nКупи слоника! ";
        cin >> answer;
    }while (answer != 'y');
}
```

Цикл с постусловием. Оператор do while. Пример 2.

Осуществить ввод последовательности чисел до первого отрицательного или равного нулю числа и найти их произведение



```
float a,p;
a=1;
p=1;
do
{
    p=p*a;
    printf("\n Input number");
    scanf("%f",&a);
}
while(a>0);
printf("\n p=%f",p);
```

Оператор do-while. Пример 3 - вычисление квадратного корня с заданной точностью

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

```
#include <math.h>
```

```
int main() {
```

```
    double X, Eps;
```

```
    double Yp, Y = 1;
```

```
    printf("Введите аргумент и точность: ");
```

```
    scanf("%lf%lf", &X, &Eps);
```

```
    do{
```

```
        Yp = Y;
```

```
        Y = (Yp + X/Yp)/2;
```

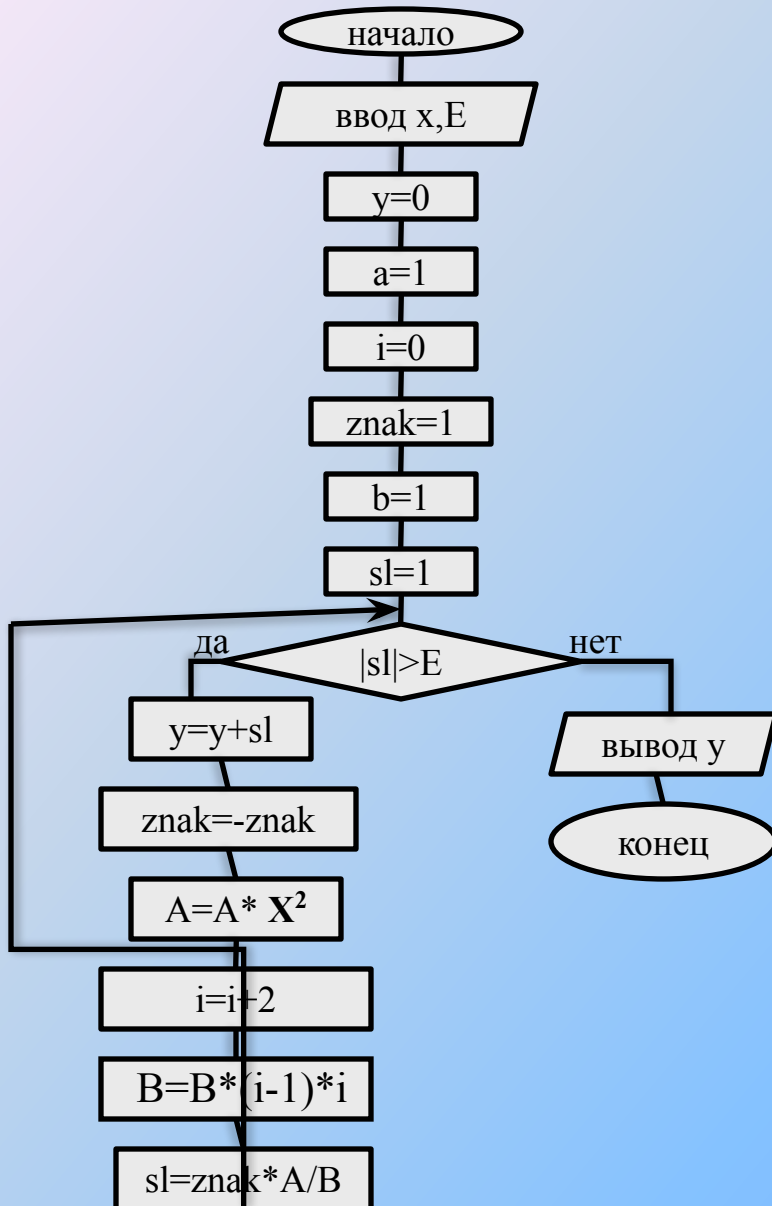
```
    }while (fabs(Y - Yp) >= Eps);
```

```
    printf("\n %lf %lf", X, Y);
```

```
}
```

$$y_n = \frac{1}{2} * (y_{n-1} + x/y_{n-1})$$

Пример 4. Вычисление суммы бесконечного ряда с заданной ТОЧНОСТЬЮ.



$$Sl = \text{znak} * A / B$$

A-числитель

B-знаменатель

Закон изменения A: $A = A * X^2$

Изменение B: $1 * 2$

$$1 * 2 * 3 * 4$$

$$1 * 2 * 3 * 4 * 5 * 6$$

...

$$B = B * (i-1) * i$$

Изменение знака: $\text{znak} = -\text{znak}$;

Количество
множителей

$$i=2$$

$$i=4$$

$$i=6$$

Вычислить сумму бесконечного ряда с заданной точностью E.

$$y = \cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

Фрагмент программы вычисления суммы ряда с заданной точностью

```
float e,x,y,a,b,sl;  
int znak,i;  
printf(“\n Input x,e:”);  
scanf(“%f%f”,&x,&e);  
y=i=0;  
a=znak=b=sl=1;  
while (abs(sl)>e)  
{  
    y+=sl;  
    znak=-znak;  
    a*=x*x;  
    i+=2;  
    b*=(i-1)*i;  
    sl=znak*a/b;  
}  
printf(“\n y=%8.2f”,y);
```

Оператор цикла **for**

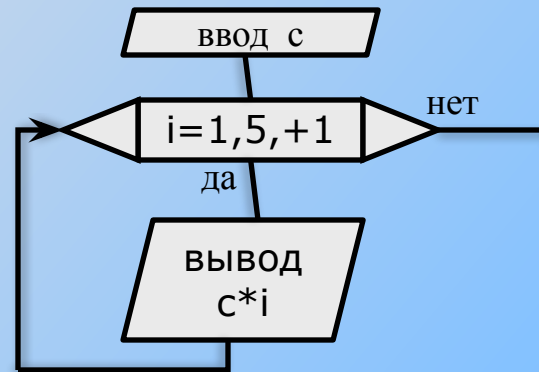
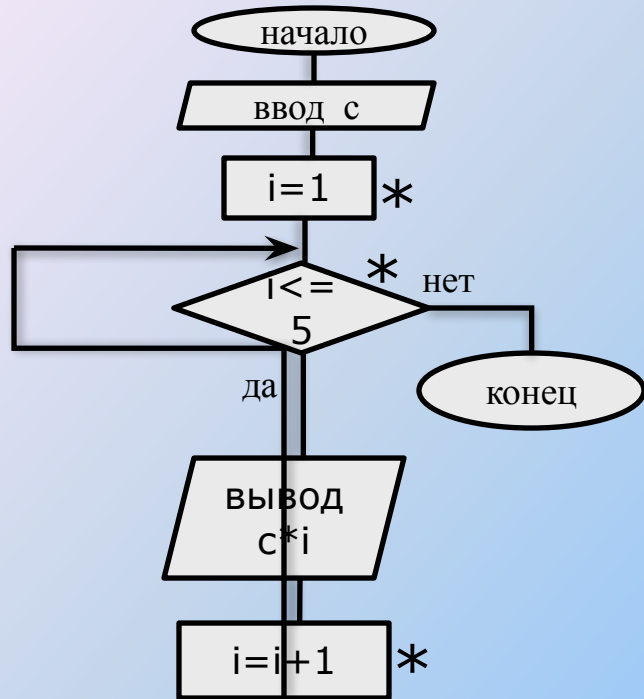
for (инициализация; выражение; модификация) оператор

Сумма чисел от 1 до 100:

```
for (int i = 1, s = 0; i <= 100; i++) s += i;
```

```
for ( int i = 1; i < 6; ++i )  
    printf( " %d ", i );  
printf( "вышел зайчик погулять" );
```

Оператор for. Пример 2.



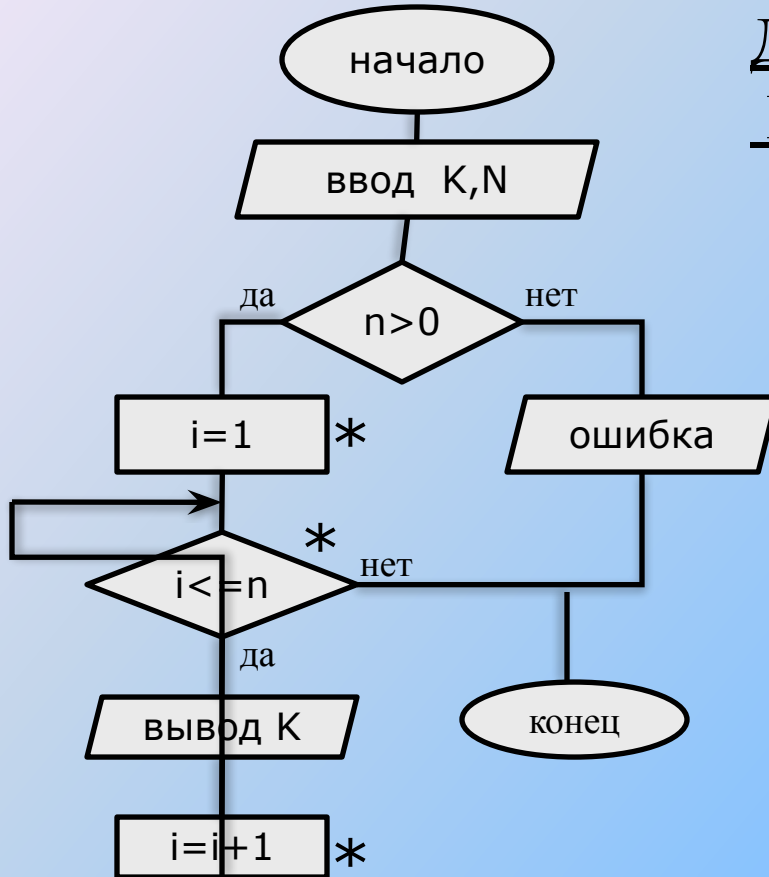
```
float c;  
int i;  
printf("\n Input c");  
scanf("%f",&c);  
for(i=1;i<=5;i++)  
    printf("\n %8.2f",c*i);
```

Известна цена 1кг конфет. Вывести стоимость 1,2,...,5 кг конфет.

Оператор for. Пример 3.

Даны 2 целых числа K и N.

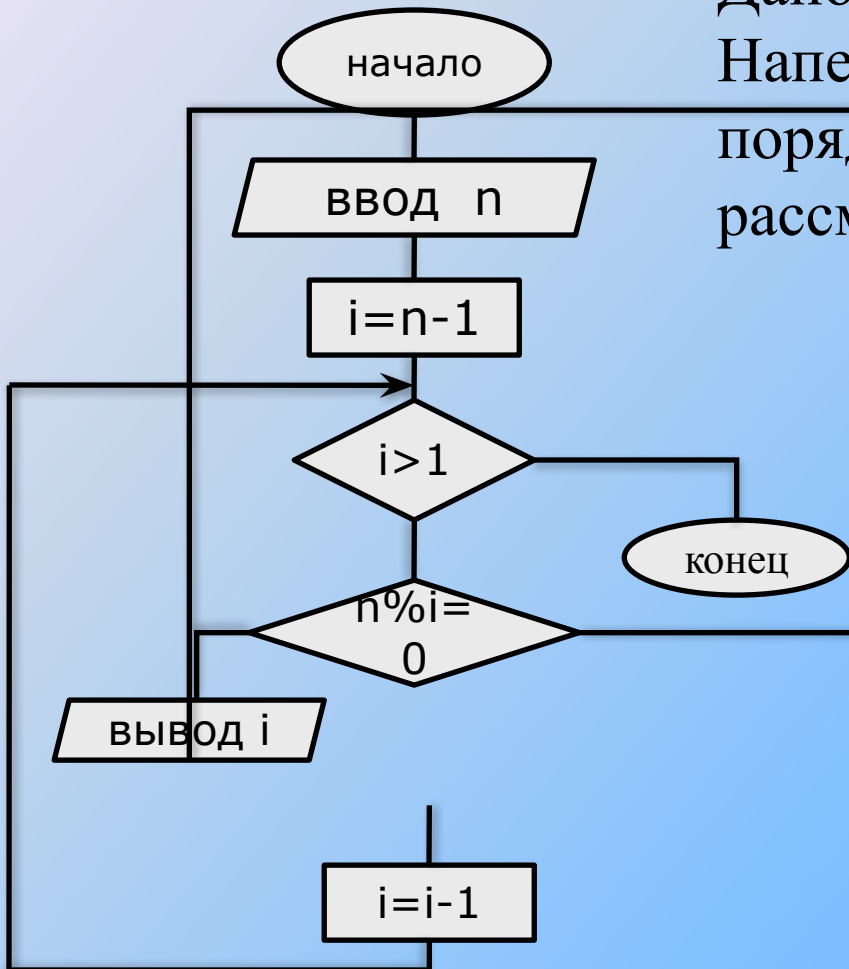
Вывести N раз число K (если N>0).



```
float k;
int n;
printf("\n K=");
scanf("%f",&k);
printf("\n N=");
scanf("%d",&n);
if (n>0)
{
    for (i=1; i<=n; i++)
        printf("\n%8.2f",k);
}
else printf("Error!");
```

Оператор for. Пример 4.

Дано натуральное число n .
Напечатать все делители числа в порядке убывания (само число и 1 не рассматривать).



```
int i,n;
printf("Input n:");
scanf("%d",&n);
for (i=n-1;i>1;i--)
    if (n%i==0) printf("\n %d",i)
```

Операторы передачи управления

- оператор безусловного перехода goto;

goto метка;

В теле той же функции должна присутствовать ровно одна конструкция вида:
метка: оператор;

- оператор выхода из цикла break;
- оператор перехода к следующей итерации цикла continue;
- оператор возврата из функции return;
- оператор генерации исключения throw

```
// ищет сумму чисел вводимых с клавиатуры до тех пор,  
// пока не будет введено 100 чисел или 0  
for(s=0, i=1; i<100;i++)  
{  
cin>>x;  
if( x==0) break; // если ввели 0, то суммирование заканчивается  
s+=x;  
}
```

```
//ищет количество и сумму положительных чисел  
for( k=0,s=0,x=1;x!=0;)  
{  
cin>>x;  
if (x<=0) continue;  
k++;s+=x;  
}
```

```
//использование goto  
int k;  
goto m;  
...  
{ int a=3,b=4;  
k=a+b;  
m: int c=k+1;  
...  
}
```