

1 курс

**ПЛАН – КОНСПЕКТ**  
проведения занятий (лекционных, практического № 21) по дисциплине  
«Информатика»

**Раздел 4. «Основы алгоритмизации и программирования.»**

**Тема 4.1: «Общие принципы построения базовых  
алгоритмических структур в среде программирования.»**

**части 9,10**

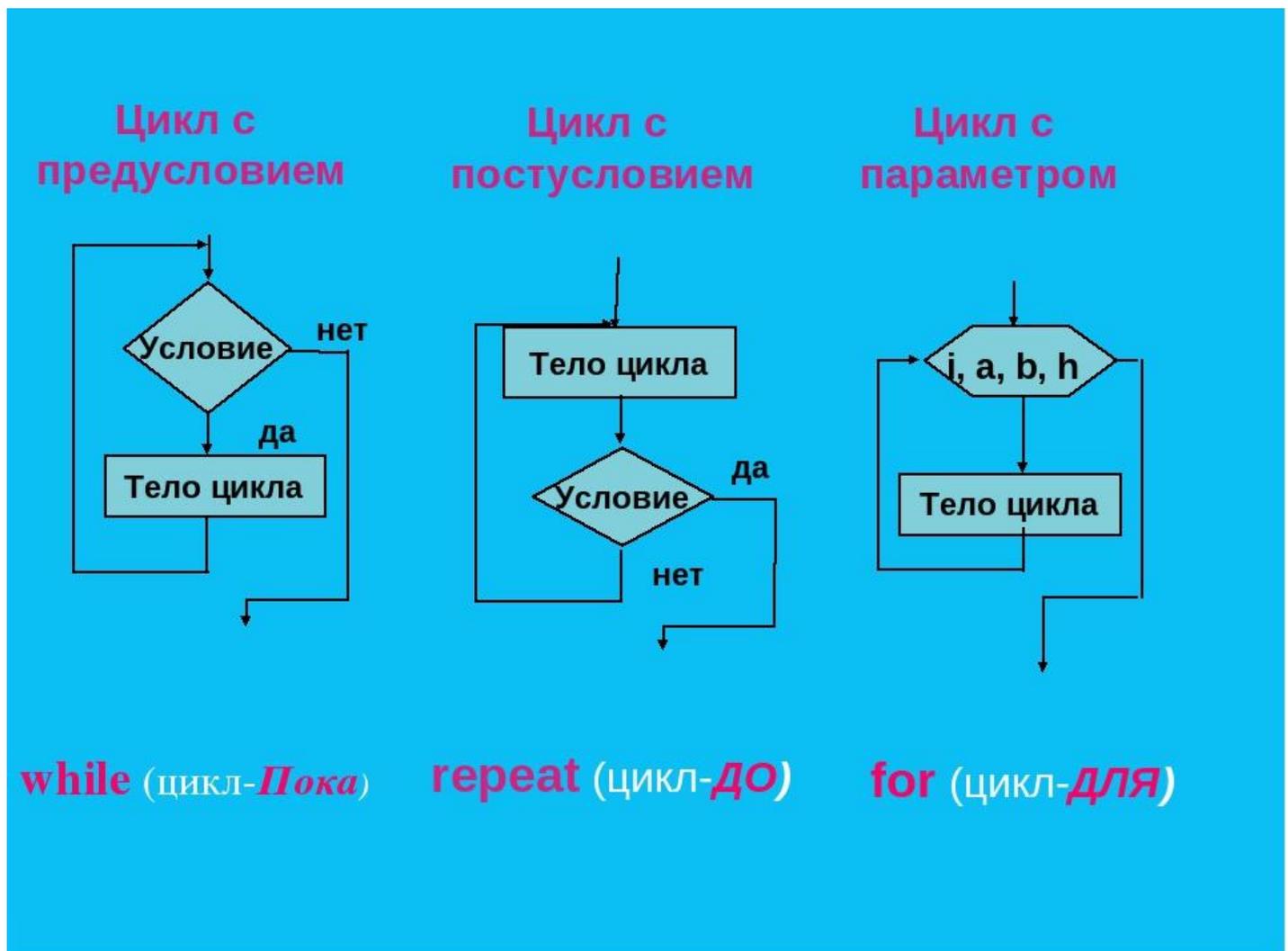
Подготовил: преподаватель  
В.Н. Борисов

## Вопросы занятий:

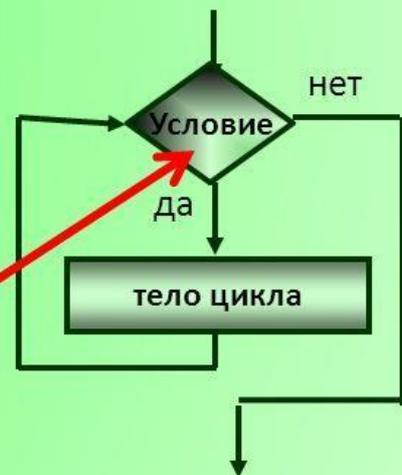
1. Циклические операторы. Циклы с предусловием, с постусловием, с параметром.
2. Составление программы по блок-схеме.
3. Аналитический расчёт результатов выполнения циклических программ.
4. Программирование цикла с предусловием (практическое занятие №21, теоретическая часть).
5. Составление программ с использованием циклов с предусловием, с постусловием, с параметром.
6. Сочетание цикла и разветвления.
7. Вложенные циклы.

Время проведения занятий (лекционных, практического) – 6 часов.

Первый вопрос: Циклические операторы. Циклы с предусловием, с постусловием, с параметром.



## Цикл с предусловием



...

```

While условие do
  Begin
    тело цикла
  End;

```

...

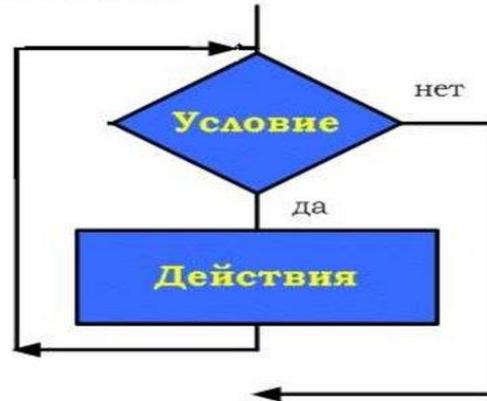
- Тело цикла выполняется, пока условие **ИСТИННО**.
- Чтобы не было заикливания, переменные, входящие в условие, должны изменяться в теле цикла.

MyShared

[Назад](#)

## Цикл с предусловием

Сначала проверяется условие. Если оно истинно, то выполняется тело цикла. Если условие становится ложным, то тело цикла не выполняется, а выполняется следующий за END оператор. Если условие с самого начала ложно, то тело цикла не выполнится ни разу.



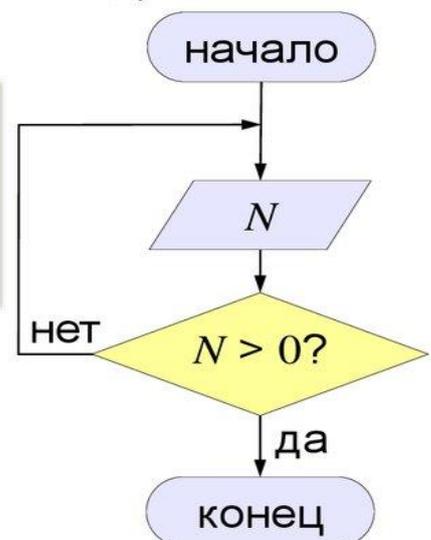
```
WHILE (ПОКА УСЛОВИЕ ИСТИННО) DO
begin
    действие1;
    действие2;
end
```

## Цикл с постусловием

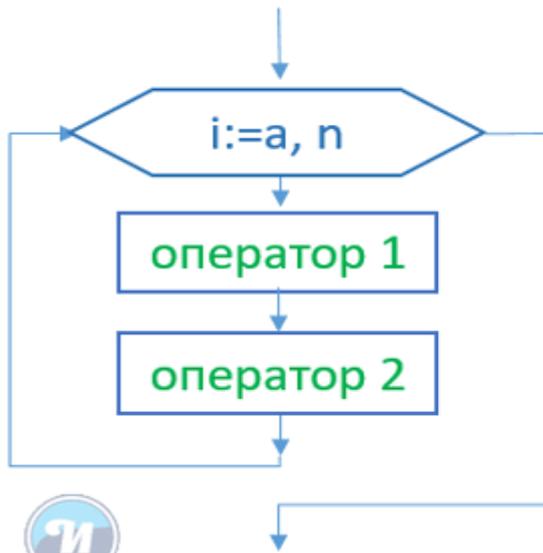
- условие проверяется **после** завершения очередного шага цикла
- цикл всегда выполняется хотя бы один раз
- как только условие становится **ИСТИННЫМ**, работа цикла заканчивается

```
repeat
    write ('Введите N>0: ');
    read (N);
until N > 0;
```

условие **окончания**  
работы цикла



## Цикл с параметром Pascal (цикл-для)



$i$  – параметр цикла

$a$  – начальное значение параметра

$n$  – конечное значение параметра

**Счётчик с шагом 1:**

for  $i:=a$  to  $n$  do оператор 1;

**Счётчик с шагом -1:**

for  $i:=a$  downto  $n$  do оператор 1;

**Использование 2 и более операторов**

for  $i:=a$  to  $n$  do

begin

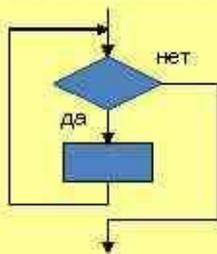
оператор 1;

оператор 2;

end;

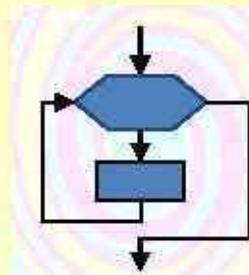
## Циклические операторы на Паскале

Оператор цикла с предусловием

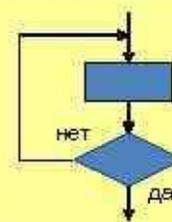


WHILE **условие** DO  
**оператор**

Оператор цикла с параметром



Оператор цикла с постусловием



REPEAT  
**оператор**  
UNTIL **условие**

параметр цикла ↑	FOR	<b>переменная</b>	:=	<b>выражение 1</b>	TO	<b>выражение 2</b>	DO	<b>оператор</b>
параметр цикла ↓	FOR	<b>переменная</b>	:=	<b>выражение 1</b>	DOWNTO	<b>выражение 2</b>	DO	<b>оператор</b>
		Имя параметра (счетчика) цикла		Начальное значение счетчика цикла		Конечное значение счетчика цикла		

Цикл (FOR ... TO ... DO ...) работает следующим образом: параметру цикла присваивается значение выражения 1, потом это значение сравнивается со значением выражения 2. Если условие выполнения цикла истинно (значение счетчика меньше значения выражения 2), то выполняется тело цикла. Далее значение параметра цикла автоматически увеличивается на единицу, снова сравнивается со значением выражения 2, если условие истинно, то опять выполняется тело цикла. Цикл прекращается, когда значение параметра цикла станет больше значения выражения 2.

При описании циклов с постусловием необходимо принимать во внимание следующее:

- перед первым выполнением цикла условие его окончания (или продолжения) должно быть определено;
- тело цикла должно содержать хотя бы один оператор, влияющий на условие окончания (продолжения), иначе цикл будет бесконечным;
- условие окончания цикла должно быть в результате выполнено.
- Для досрочного выхода из цикла используют в QBasic операторы EXIT DO (LOOP), в Turbo Pascal 7.0 функции EXIT, BREAK.



Второй вопрос: Составление программы по блок-схеме.

## Цикл с постусловием Repeat ... until

Решение задачи о выводе 10 целых чисел на экран с использованием цикла *Repeat...until*:



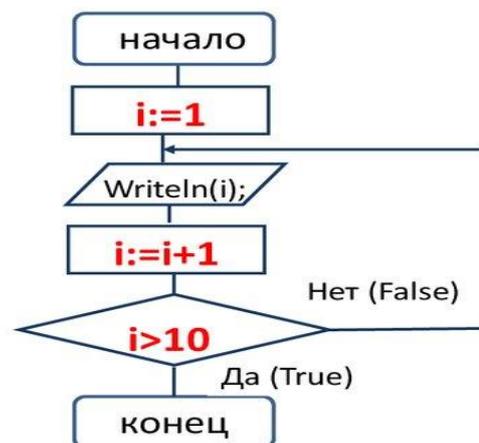
### Программа

```
var i: integer; {счетчик}
Begin
  i := 1; {начальное значение}

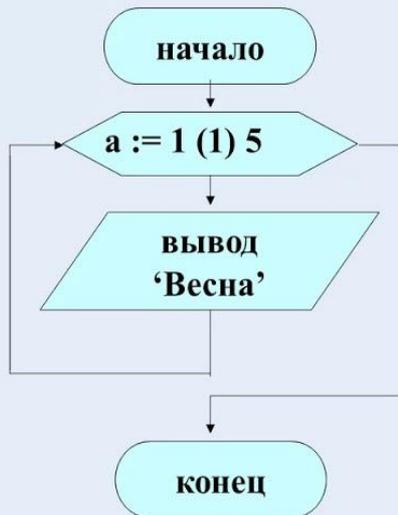
  Repeat
    Writeln(i);
    i := i + 1
  Until i > 10
End.
```



### Блок-схема алгоритма

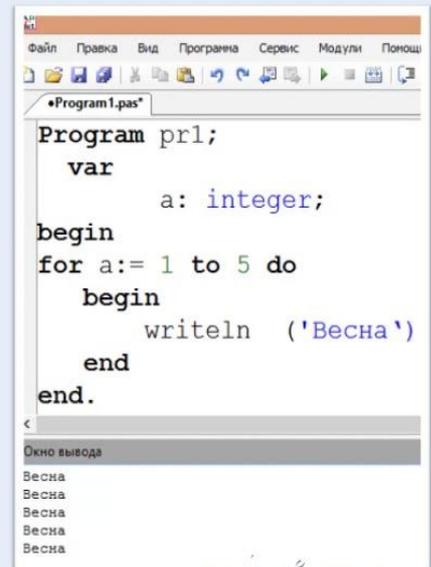


## 1. Вывести на экран 5 раз слово «Весна».



```

Program pr1;
var
  a: integer;
begin
  for a:= 1 to 5 do
  begin
    writeln ('Весна')
  end
end
end.
  
```



## Пример использования циклов

Необходимо вывести на экран цифры от '0' до '9', используя цикл с предусловием и цикл с постусловием.

**Цикл с предусловием:**

```

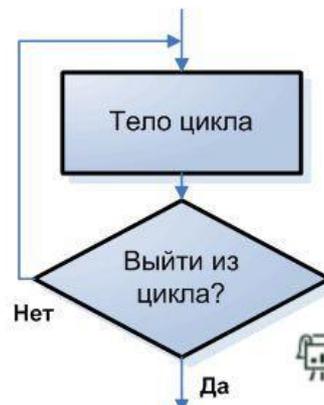
mov dl, '0'
mov ah, 02
cikl: cmp dl, '9'
      ja end_cikl
      int 21h
      inc dl
      jmp cikl
end_cikl:
  
```



**Цикл с постусловием:**

```

mov dl, '0'
mov ah, 02
cikl:
      int 21h
      inc dl
      cmp dl, '9'
      jbe cikl
  
```

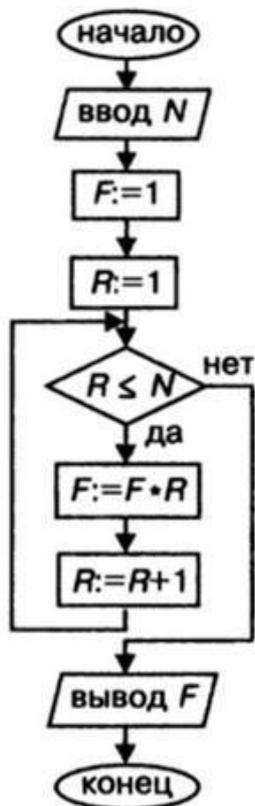


Третий вопрос: Аналитический расчёт результатов выполнения циклических программ.

Четвёртый вопрос: Программирование цикла с предусловием (практическое занятие №21, теоретическая часть).

## Алгоритм нахождения факториала

Блок-схема :



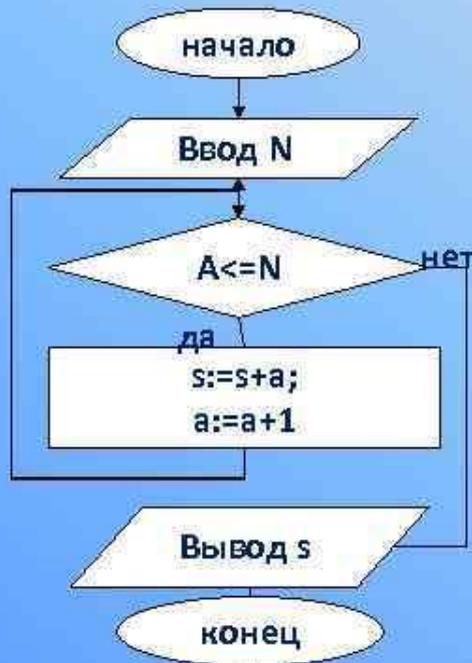
Программа на Паскале:

```

Program Words;
var
N, R, F:integer;
Begin
Write('Введите количество букв:');
Readln(N);
F:=1; R:=1;
While R<=N do
begin
F:=F*R;
R:=R+1;
end;
Writeln('Из ',N,' букв составим ',F,' слов');
End.
  
```

### Цикл с предусловием (цикл – пока)

Задача. Вычислить сумму натурального ряда чисел от 1 до N.

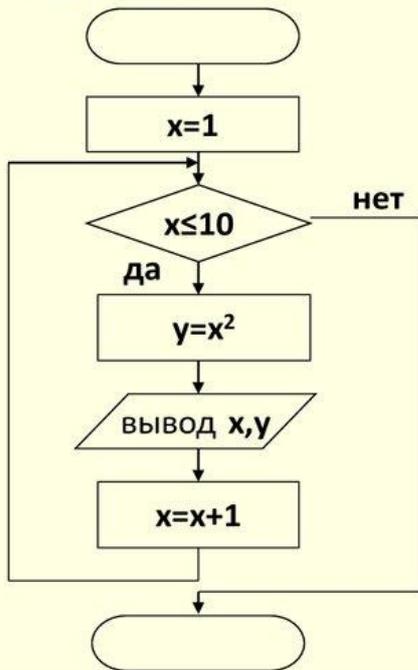


```

Program primer;
Var a, s, n: integer;
Begin
Write('введите N');
Readln(N); S:=0;
While a<=N do
begin
s:=s+a;
a:=a+1;
End;
Writeln('результат
суммирования равен', s);
Readln;
end.
  
```

# Задача 1 (табулирование функции)

Вычислить значения функции  $y=x^2$  на отрезке  $[1,10]$  с шагом 1

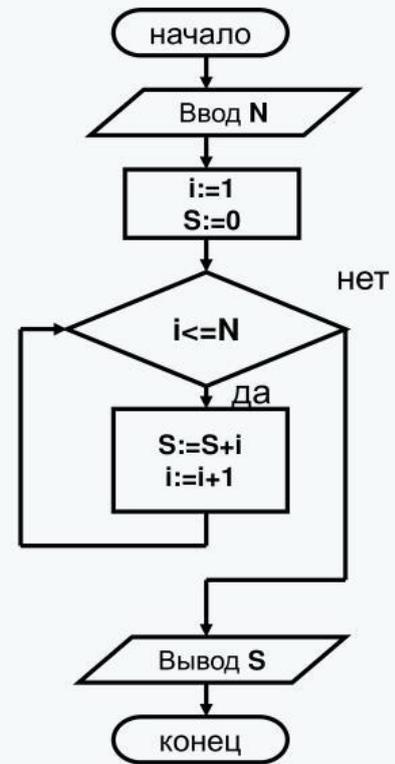


```
Program test1;
Var x,y: integer;
Begin
x := 1;
  While x<=10 do
    Begin
      y:= sqr(x);
      Writeln ('x=',x:2,' y=',y);
      x:= x+1;
    End;
  End.
End.
```

## Решение задач (цикл типа пока)

**Задача 1.** Вычислить сумму натурального ряда чисел от 1 до  $N$ .

```
Program summa_1;  
Var i,N,S:integer;  
begin  
  Write('N=');  
  Readln(N);  
  i:=1; S:=0;  
  While i<=N do  
  begin  
    S:=S+i;  
    i:=i+1  
  end;  
  Writeln('Summa=',S)  
end.
```



Вычислить значение  $y = \sqrt{x}$ , если  $x$  изменяется от 5 до 15 с шагом 0,5.

Цикл с предусловием

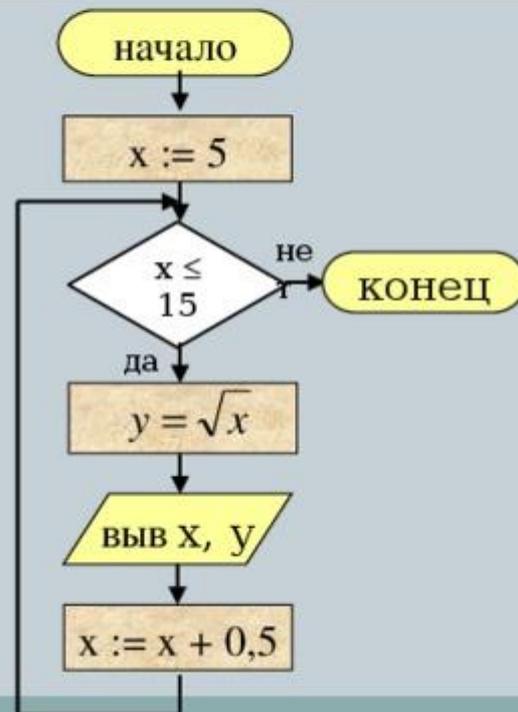
Блок-схема

### Программа

```

program Lr9;
uses Crt;
var x, y: real;
Begin
clrscr;
x:= 5;
While x<=15 do
begin
y = sqrt(x);
Writeln ('Pri x= ', x:0:1, ' y= ',
y:0:3);
x:= x + 0.5;
end;
readkey;
End.

```



**Пятый вопрос: Составление программ с использованием циклов с предусловием, с постусловием, с параметром.**

Пример составления схем алгоритмов, программ с использованием циклов с предусловием, с постусловием, с параметром рассмотрены также при рассмотрении вопросов 2,4 данных занятий.

**Пример:** Используя оператор **repeat ... until** вычислить сумму  
 $S=1+1/2+1/3+\dots+1/n$



Цикл с постусловием

```

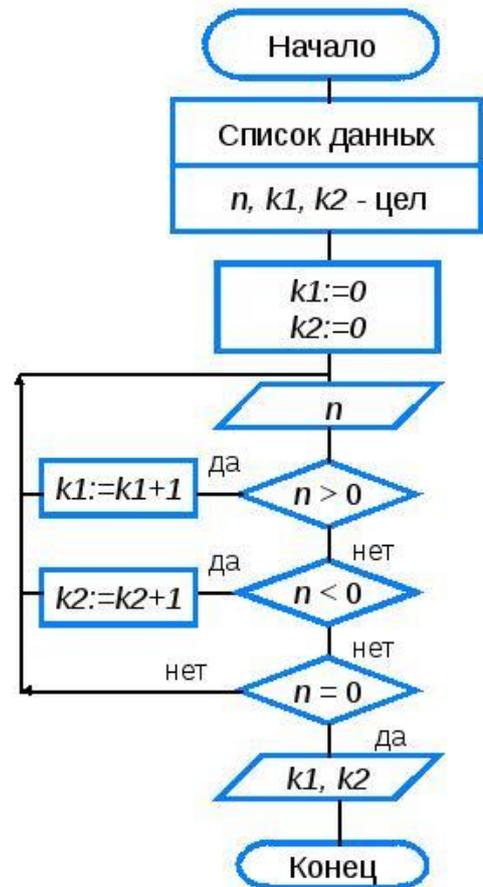
program rep_unt;
var S: real;
    n,i:integer;
begin
  writeln('Введите n->');
  readln(n);
  S:=0;
  i:=1;
  repeat
    S:=S+1/i; - цикл
    i:=i+1; - изм.переменной
  until i>n; - условие
  writeln('S=',S:10:6);
  readln;
end.
  
```

**Шестой вопрос:** Сочетание цикла и разветвления.

```

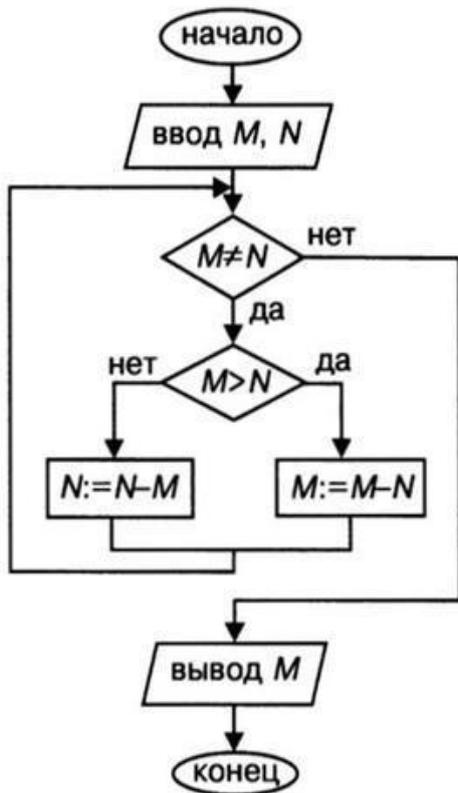
program n_17;
  var n, k1, k2: integer;
begin
  k1:=0;
  k2:=0;
  repeat
    write ('Введите целое
число>>');
    readln (n);
    if n>0 then k1:=k1+1;
    if n<0 then k2:=k2+1;
  until n=0;
  writeln ('Введено:');
  writeln ('положительных
чисел - ', k1);
  writeln ('отрицательных
чисел - ', k2)
end.

```



## Алгоритм Евклида

Блок-схема :



Программа на Паскале:

```

Program NCD;
var
  M, N:integer;
Begin
  Write('Введите M и N:');
  Readln(M, N);
  While M<>N do
  begin
    If M>N then
      M:=M-N
    else
      N:=N-M
    end;
  Writeln('НСД=',M);
End.
  
```

## ЗАДАЧА

ВЫЧИСЛИТЬ СУММУ НАТУРАЛЬНОГО РЯДА ОТ 50 ДО 100

### Цикл с предусловием

```

Program SNW;
VAR K,S:integer;
BEGIN

K:=50;
S:=0;
WHILE K<=100 DO
  BEGIN
    S:=S+K;
    K:=K+1;
  END;
WRITE(S);
END.

```

### Цикл с постусловием

```

Program SNR;
VAR K,S:integer;
BEGIN

K:=50;
S:=0;
REPEAT

S:=S+K;
K:=K+1;
UNTIL K>100;
WRITE(S);
END.

```

### Цикл с параметром

```

Program SNF;
VAR K,S:integer;
BEGIN

S:=0;
FOR K=50 TO 100 DO

S:=S+K;

WRITE(S);
END.

```

Седьмой вопрос: Вложенные циклы.

# Вложенные циклы

**Вложенный цикл** – цикл, содержащий в себе несколько простых циклов. Различают внешний и внутренние циклы. Внутренние циклы должны записываться таким образом, чтобы они полностью находились внутри других циклов.

**Пример 1.** (Печатает 10 строк пар чисел)

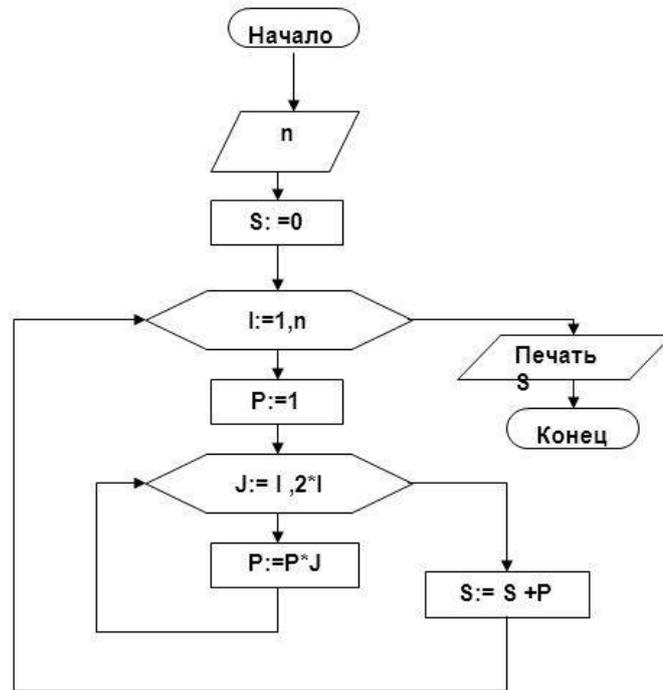
```
Program Primer1;
Var l,k : integer;
Begin
  for i:=1 to 10 do
  begin
    for k:=1 to 3 do
      writeln ('i=', l , 'k=' , k);
    writeln;
  end;
  readln;
end.
```

**Пример 2.** Найти все такие тройки натуральных чисел  $x, y, z$  из интервала от 1 до 20, для которых выполняется равенство:  $x^2 - y = z^2$ .

```
Program Primer2;
Uses crt;
Var x, y, z : integer;
Begin
  clrscr;
  for x:=1 to 20 do
    for y:=1 to 20 do
      for z:=1 to 20 do
        if x*x - y = z*z then
          writeln (x, ' ', y, ' ', z);
      end;
    end;
  readln;
end.
```

# TURBO PASCAL

## ВЛОЖЕННЫЕ ЦИКЛЫ





# Вложенные циклы

Например.

Даны натуральные числа  $n$  и  $k$ . Составить программу вычисления выражения  $1k+2k+\dots+nk$ .

Решение

Для вычисления указанной суммы целесообразно организовать цикл с параметром  $i$ , в котором, во-первых, вычислялось бы очередное значение  $y=ik$  и, во-вторых, осуществлялось бы накопление суммы прибавлением полученного слагаемого к сумме всех предшествующих ( $s= s+y$ ).



## Вложенные циклы

```
Program Example_51;  
Var n, k, y, i, s, m: Integer;  
Begin  
  Writeln('Введите n и k ');  
  Readln(n,k);  
  s:=0;  
  For i:=1 To n Do  
    Begin  
      y:=1;  
      For m:=1 To k Do y:=y*i;  
      {нахождение степени k числа i}  
      s:=s+y;  
    End;      Writeln('Ответ: ',s);      End.
```

Для решения задачи потребовалось организовать два цикла, один из которых пришлось поместить внутрь другого. Такие конструкции называют *вложенными циклами*.