

---

# **Проверочная работа «Цикл с предусловием»**



---

□ I. Запишите конструкцию цикла с предусловием.

В каких случаях применяется цикл с предусловием?

---



---

**2. Запишите правильную структуру последовательности операторов.**

**a:=2; b:=1; while a+b<8 do begin a:=a+1;  
b:=b+1 end; s:=a+b.**

**Сколько раз будет повторен цикл и какими будут значения переменных a, b, s после его завершения?**

---



---

**3. Запишите правильную структуру последовательности операторов. Какими будут значения переменных a и b после выполнения операторов?**

**a:=1; b:=1; while a<=8 do a:=a+1;  
b:=b+1.**



---

4. Дана последовательность операторов, вычисляющих факториал  $f$  числа  $n$ , которая содержит пять ошибок. Найдите эти ошибки, предварительно записав операторы в правильном виде.

$k:=1;f:=0;$

while  $k < n$  do  $f=f*k$

$k:=k+1,$

*Примечание.* Факториалом натурального числа называется произведение всех натуральных чисел от 1 до этого числа, т. е.  $n! = 1*2*3*...*(n-2)*(n-1)*n$ .



---

**5.** Найдите и исправьте ошибки в следующем фрагменте программы, определяющей для заданного натурального числа  $n$  число, записанное цифрами числа  $n$  в обратном порядке.

```
p:=n;  
while p>=0 do begin  
a:=a+p mod 10;  
p:=p div 10  
end;
```

---



## Ответы:

---

`while <логическое  
выражение> do begin  
группа операторов end;`

Когда до выполнения цикла  
количество повторов неиз-  
вестно.

---



---

**2.**

**a:=2;**

**b:=1;**

**while a+b<8 do**

**begin**

**a:=a+1;**

**b:=b+1**

**end;**

**s:=a+b.**

**2 раза;**

**a = 5, b = 3, s = 8.**



---

**3.** a:=2; b:=3;

while a<=7 do

a:=a+1;

b:=b+1;

a = 8, b = 4.

**4.** k:=1; f:=1;

while k<=n do f=f\*k;

k:=k+1;

end;

---



---

**5.** read(n);

p:=n; a:=0;

while p>=0 do

begin

a:=a\*10+p mod 10;

p:=p div 10

end;



- 
- Вы уже умеете организовать цикл при помощи оператора **while**.
  - Напомним, что при выполнении этого оператора компьютер вычисляет значение условия.
- 
- 

- 
- Если условие истинно, то исполнительная часть оператора **while** будет выполняться до тех пор, пока это условие не примет значение **false**.
  - Если значение условия есть **false** в самом начале, то исполнительная часть оператора **while** вообще не будет выполняться.
- 
- 

---

□ Иногда при решении задач возникает необходимость выполнить тело цикла хотя бы один раз, а потом исследовать условие, повторять ли его еще раз. Эту задачу выполнит другой вид цикла **Repeat**.

---



- 
- **repeat** повторяй операторы
  - **until** <условие>; до тех пор, пока условие не будет верным
  - Есть небольшое отличие в организации цикла **repeat** по сравнению с **while**: для выполнения в цикле **repeat** нескольких операторов не следует помещать эти операторы в операторные скобки **begin...end**. Резервированные слова **repeat** и **until** действуют как операторные скобки.
- 



---

□ Конструкция **repeat... until** работает аналогично циклу **while**. Различие заключается в том, что цикл **while** проверяет условие до выполнения действий, в то время как **repeat** проверяет условие после выполнения действий, это гарантирует хотя бы одно выполнение действий до завершения цикла.

---



Например:

---

repeat

read (Number);

Sum:=Sum+Number;

until Number=- 1

repeat

i:= i+ 1;

writeln (Sqr(i))

until Number=-1

---



---

# Практическая работа

---



## **Задание 1.**

---

**Определите, является ли данное число простым.**

**Примечание. Простым называется число, которое не имеет делителей, кроме 1 и самого себя.**

---



# Решение:

```
Var i, Number : integer; {возможный делитель и  
исследуемое число}  
Begin  
writeln ('Какое число должно быть проверено?');  
read (Number);  
i:=1;  
repeat  
i:=i+1;  
until Number mod i=0;  
if Number=i then writeln(Number,'является простым')  
else writeln(Number,'делится на',i);  
End.
```

## Задание 2. Выведите на экран строку из звездочек

---

Решение:

```
Var i, n: integer;  
begin  
write('Количество знаков:'); readln(n);  
i:=1;  
repeat  
write(' (*)');  
i:=i+1  
until i>n;  
end.
```

---



**Задание 3. Выведите квадраты натуральных чисел, не превосходящих 50.**

---

*Решение:*

```
var x, n: integer;  
begin  
  write('Enter n:'); readln(n);  
  x:=1;  
  repeat  
    y:=x*x;  
    if y<n then write(y, ' ');  
    x:=x+1;  
  until y>n;  
end.
```

---

## Задание 4. Найдите НОД двух чисел

---

*Решение:*

```
Var x, y: integer;
```

```
Begin
```

```
Writeln ('Введите два числа'); readln(x, y);
```

```
Repeat
```

```
If  $x > y$  then  $x := x \bmod y$  else  $y := y \bmod x$ ;
```

```
Until (x=0) Or (y=0);
```

```
Writeln ('НОД=', x+y));
```

```
End.
```

---



## Вывод

---

- Как программируется цикл с постусловием на языке Паскаль?
- Для каких целей лучше его использовать?



## Домашнее задание

---

- ▣ Составить программу нахождения НОД трех чисел.  $\text{НОД}(a, b, c) = \text{НОД}(\text{НОД}(a, b), c)$ .
- ▣ Составить программу нахождения НОК двух чисел, используя формулу  $A \times B = \text{НОД}(A, B) \times \text{НОК}(A, B)$ .

**program dzl;**

Var A, B, C, nod: integer;

Begin

Readln(A,B,C);

Repeat

If  $A > B$  then  $A := A \bmod B$  else  $B := B \bmod A$ ;

Until  $(A=0)$  or  $(B=0)$ ;

Writeln('НОД=' ,A+B);

nod:=A+B;

Repeat

If  $nod > C$  then  $nod := nod \bmod C$  else  $C := C \bmod nod$ ;

Until  $(nod=0)$  or  $(C=0)$ ;

Writeln ('НОД=',nod+C);

End.



**program dz2;**

var n, m, i, nok: integer;

Begin

write('ВВЕДИТЕ ДВА ЧИСЛА');

readln(n, m);

if(m<>0) and (n<>0) then

begin

nok:=n\*m; i:=nok;

while i>0 do begin

if (i mod n=0) and (i mod m=0) then nok:=i;

i:=i-1; end;

writeln('НОК',m,'i',n,'raven',nok)

end else writeln ('Na nol delit nelzya');

end.

