

# ***Циклические алгоритмы***

Сколько циклических алгоритмов можно увидеть вокруг, если внимательно посмотреть на события: чередование времен года



ЗИМ  
А



ВЕС  
НА



ОСЕ  
НЬ



ЛЕТ  
О

посещения магазинов, школы или секции, получение за  
контрольные оценок и др.



а) Пока не сдал выпускные экзамены делай  
начало

готовь уроки;

посещай школу;

конец;

б) Пока есть желание, возможность и  
здоровье делай

начало

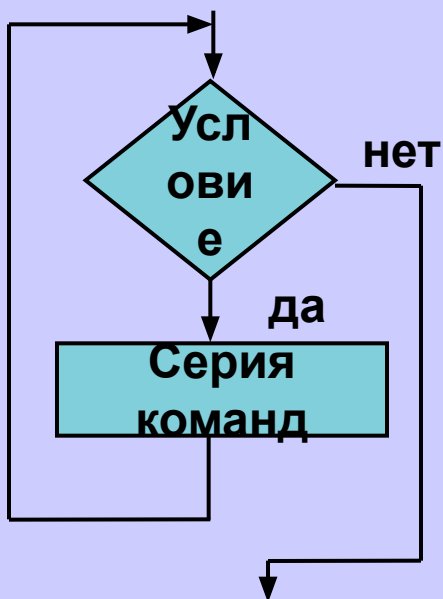
посещай спортивные занятия;

конец;

Для реализации повторяющихся действий существуют специальные алгоритмические структуры, получившие название – **ЦИКЛЫ** или команды повторения.

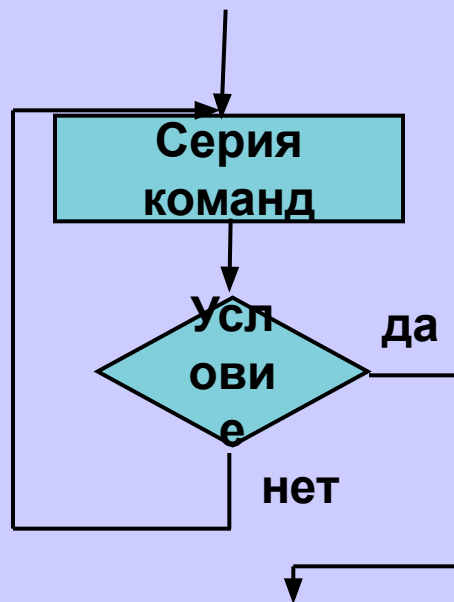
# Виды циклических алгоритмов

Цикл с  
предусловием



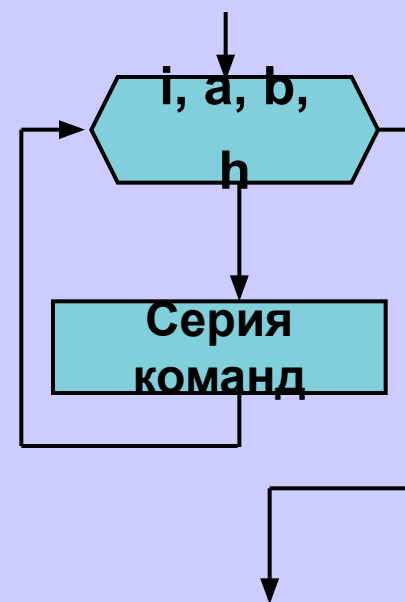
Цикл типа *Пока*

Цикл с  
постусловием



Цикл типа *до*

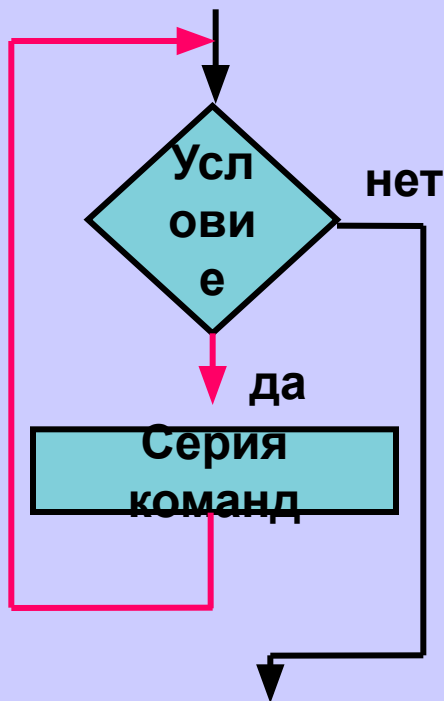
Цикл с  
параметром



Цикл типа *для*

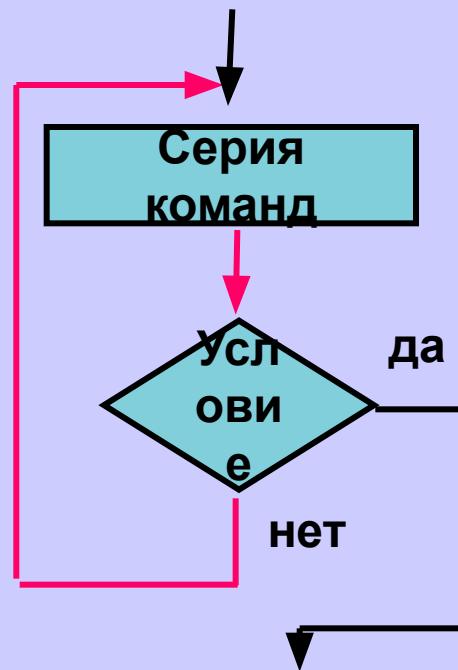
# Виды циклических алгоритмов

Цикл с  
предусловием



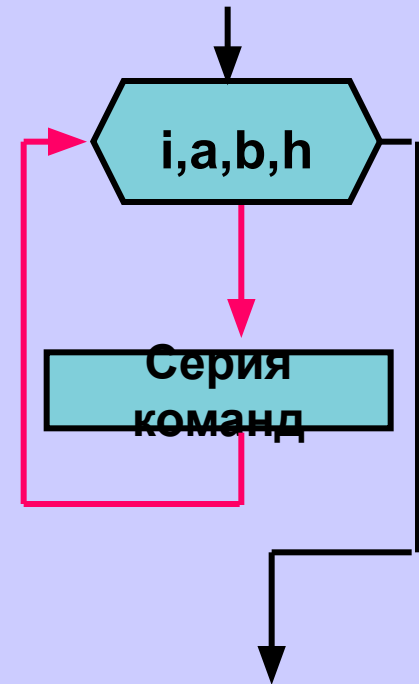
Цикл типа *Пока*

Цикл с  
постусловием



Цикл типа *До*

Цикл с  
параметром



Цикл типа *Для*

## Цикл с предусловием (типа «**ПОКА**» )

**Пока** (условие)

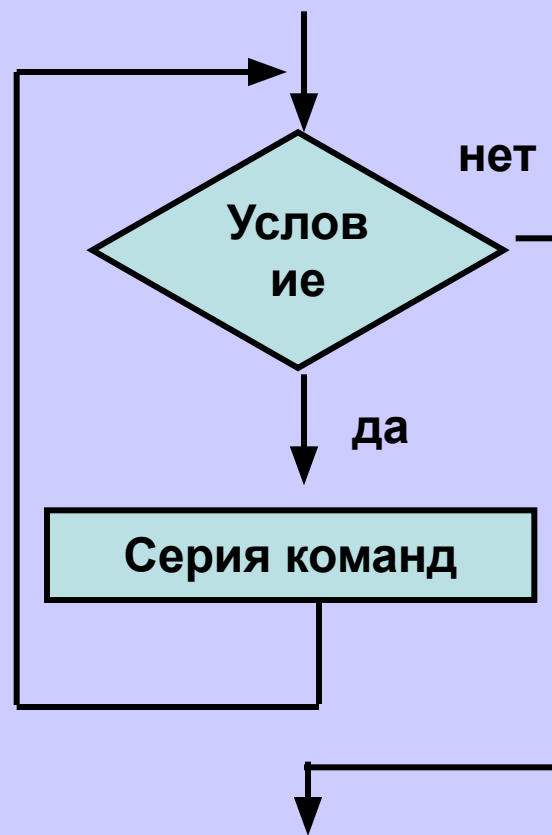
**нц**

Серия команд;

**кц**

Запись на языке  
программирования Pascal:

```
while условие do  
  begin  
    Серия команд;  
  end;
```





```
while условие do  
  begin  
  Серия команд;  
  end;
```

## Обратите внимание

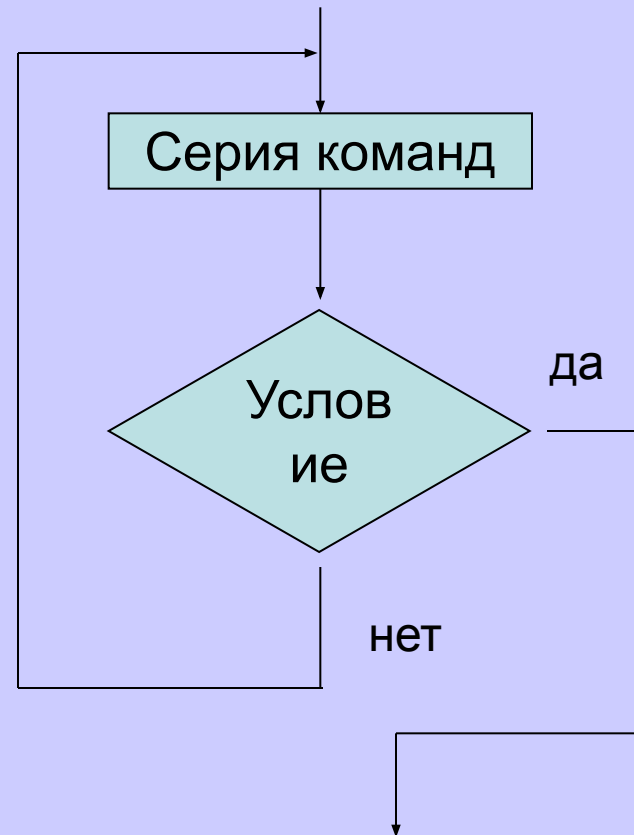
1. Цикл заканчивается, когда **условие** становится **не верным (ложным)**.
2. Если **условие** с самого начала ложно, то серия команд **не выполняется ни разу**.

## Цикл с постусловием ( типа «До» )

В алгоритмическом языке нет команды которая могла бы описать данную структуру, но ее можно выразить с помощью других команд( ветвления).

Запись на языке программирования Pascal:

```
repeat  
    Серия команд;  
until условие
```



**repeat**

Серия команд;

**until** *условие*

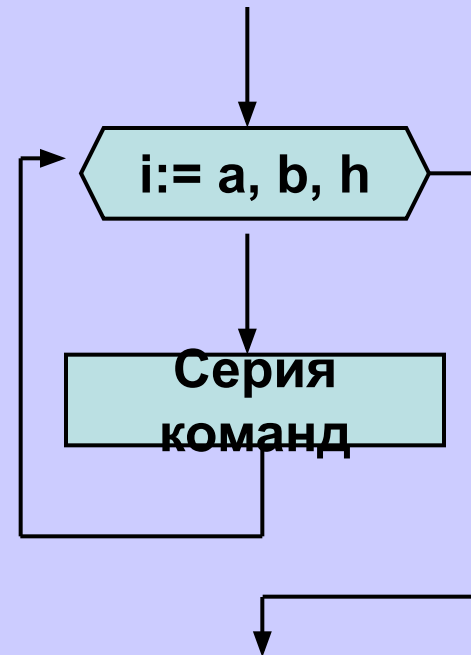
## Обратите внимание

- Серия команд между **repeat** и **until** выполняется **хотя бы один раз**.
- Цикл заканчивается когда, **условие** становится **верным (истинным)**.

Циклы типа **repeat** и **while**,  
используются в программе, если  
надо провести некоторые  
повторяющиеся вычисления  
(цикл), однако *число повторов*  
*заранее не известно и*  
*определяется самим ходом*  
*вычисления.*

# Цикл с параметром (типа «ДЛЯ»)

Для  $i$  от  $a$  до  $b$  шаг  $h$  делай  
Нц  
Серия команд;  
кц



$h = 1$

**for**  $i := a$  **to**  $b$  **do**  
begin  
Серия команд;  
**end;**

$h = -1$

**for**  $i := b$  **downto**  $a$  **do**  
begin  
Серия команд;  
**end;**

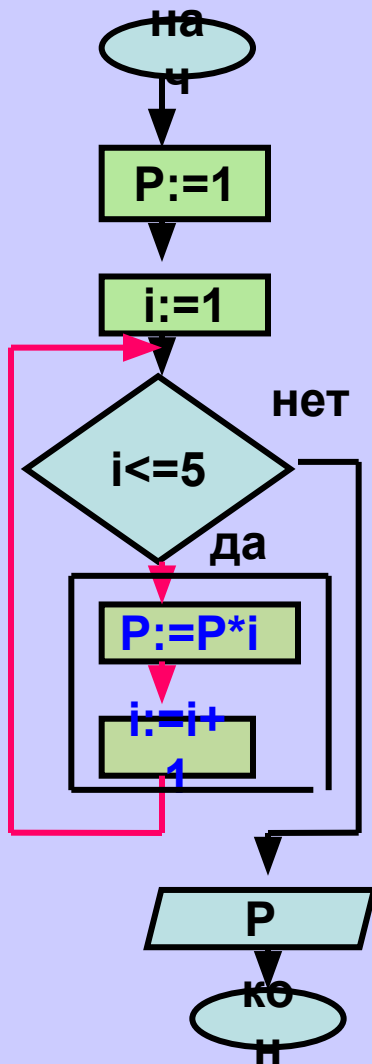
Пример:

Вычислить произведение чисел от 1 до 5 используя различные варианты цикла.

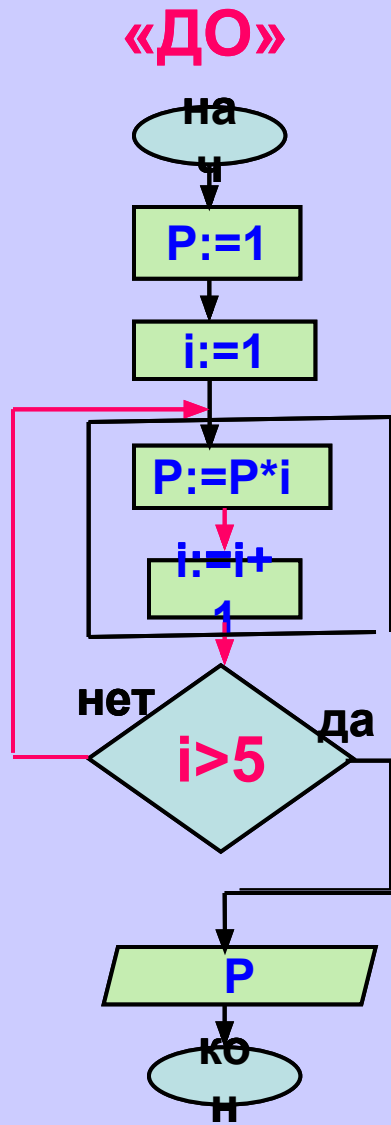
Математическая модель:

$$P = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$$

## «Пока»



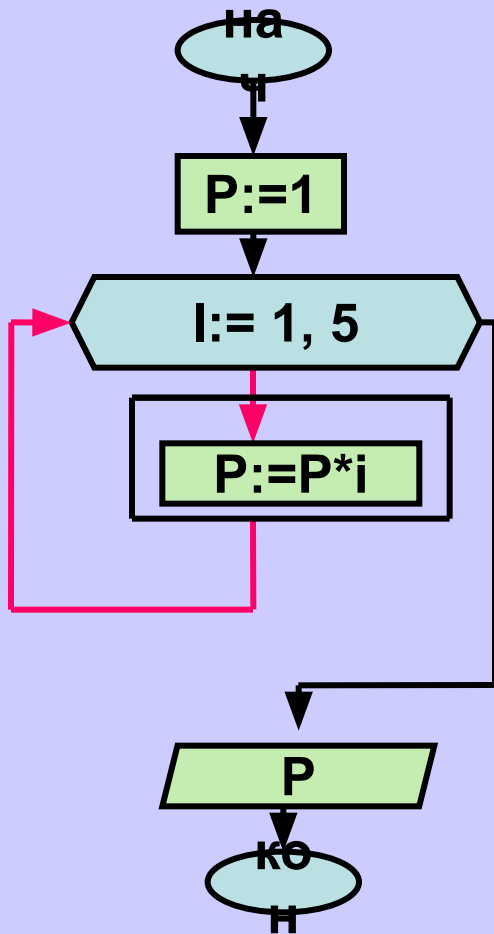
Шаг	Операция	P	i	Проверка условия
1	P:=1	1		
2	i:=1;	1	1	
3	i<=5 P:=P*i i:=i+1	1	1	1<=5, да (истина)
4	i<=5 P:=P*i i:=i+1	2	2	2<=5, да (истина)
5	i<=5 P:=P*i i:=i+1	6	3	3<=5, да (истина)
6	i<=5 P:=P*i i:=i+1	24	4	4<=5, да (истина)
7	i<=5 P:=P*i i:=i+1	120	5	5<=5, да (истина)
8	i<=5 P:=P*i i:=i+1			6<=5, нет (ложь)



Шаг	Операция	P	i	Проверка условия
1	P:=1;	1		
2	i:=1;	1	1	
3	P:=P*i; i:=i+1; i>5	1	2	2>5, нет (ложь)
4	P:=P*i i:=i+1 i>5	2	3	3>5, нет (ложь)
5	P:=P*i i:=i+1 i>5	6	4	4>5, нет (ложь)
6	P:=P*i i:=i+1 i>5	24	5	5>5, нет (ложь)
7	P:=P*i i:=i+1 i>5	120	6	6<=5, да (истина)

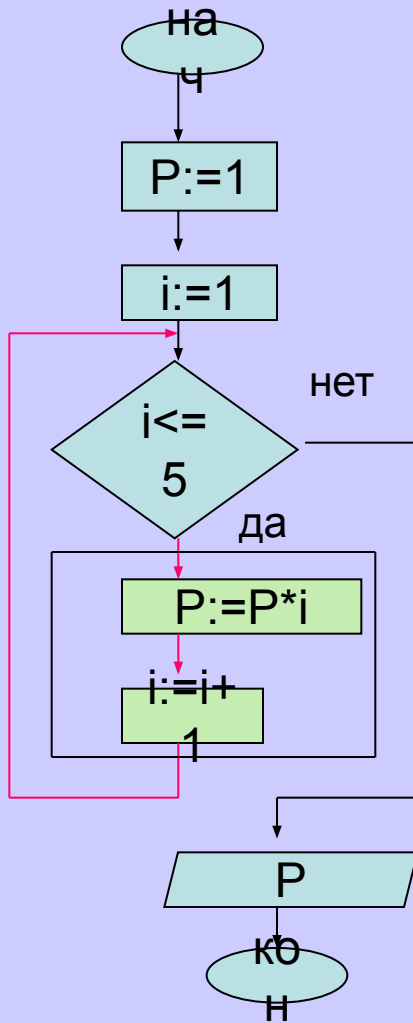


## « ДЛ Я »

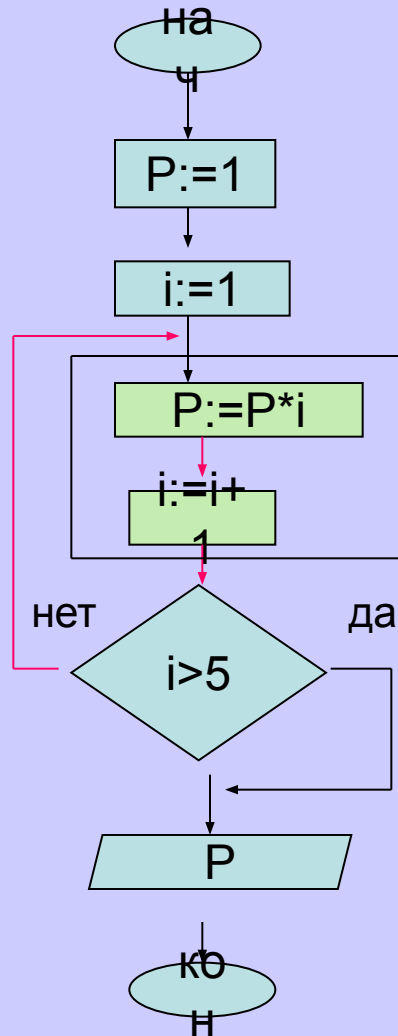


Шаг	Опера ция	P	i	Проверка условия
1	P:=1	1		
2	i:=1 P:=P*i	1	1	
3	i:=2 P:=P*i	2	2	
4	i:=3 P:=P*i	6	3	
5	i:=4 P:=P*i	24	4	
6	i:=5 P:=P*i	120	5	

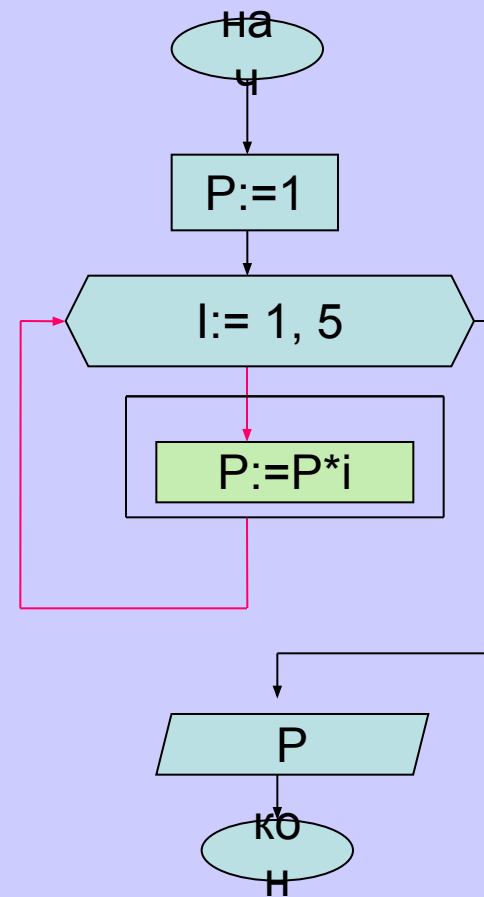
## «Пока»



## «ДО»



## « для »



## «Пока»

```
Program Pr1;  
Var i: integer;  
Begin  
P:=1;  
i:=1;  
While i<=5 do  
    begin  
        P:=P*i;  
        i:=i+1;  
    end;  
Write ('P=', P);  
end.
```

## «ДО»

```
Program Pr2;  
Var i: integer;  
Begin  
P:=1;  
i:=1;  
Repeat P:=P*i;  
    i:=i+1;  
until i>5;  
Write (' P=', P);  
end.
```

## « ДЛ Я »

```
Program Pr3;  
Var i: integer;  
Begin  
P:=1;  
For i:=1 to 5 do  
P:=P*i;  
Write ('P=', P);  
end.
```

## Задача

Вывести на экран числа от 1 до 5 в:

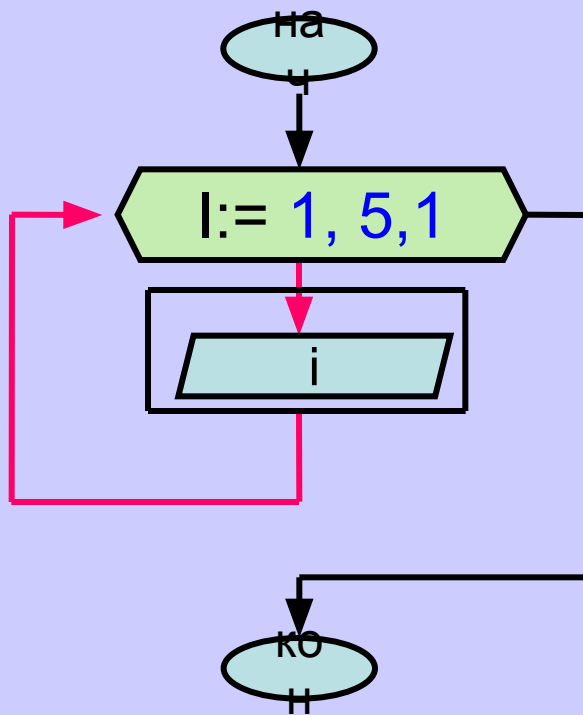
- a) прямом порядке;
- b) обратном порядке.

Математическая модель:

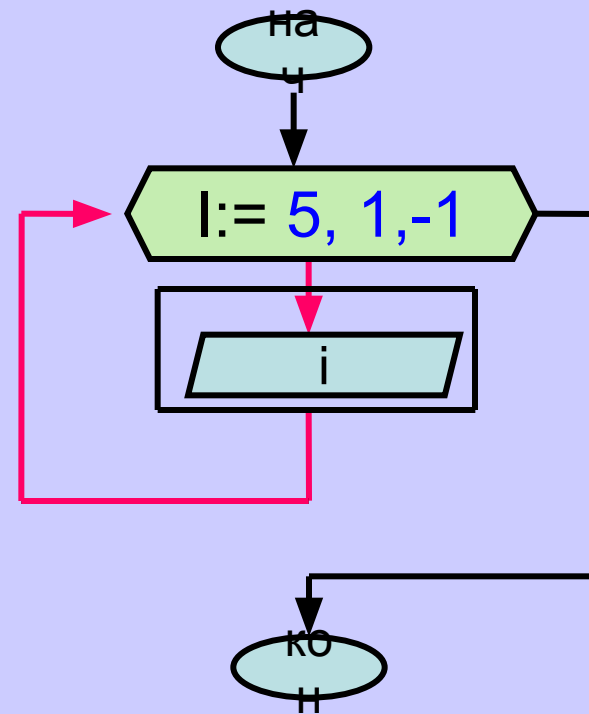
a) 1 2 3 4 5

b) 5 4 3 2 1

Для чисел в прямом порядке  $h = 1$



Для чисел в обратном порядке  $h = -1$



```
Program Pr4;  
Var i: integer;  
Begin  
For i:=1 to 5 do  
Write (i);  
end.
```

В результате на экране  
будет:

**1 2 3 4 5**

```
Program Pr5;  
Var i: integer;  
Begin  
For i:=5 downto 1  
do  
Write (i);  
end.
```

В результате на экране  
будет:

**5 4 3 2 1**

И так мы рассмотрели следующие вопросы:

1. Алгоритмическая структура цикл;
2. Виды алгоритмических структур:
  - Цикл с предусловием;
  - Цикл с постусловием;
  - Цикл с параметром;
3. Рассмотрели способы записи данных структур;
4. Разобрали примеры решения задач с помощью этих структур.