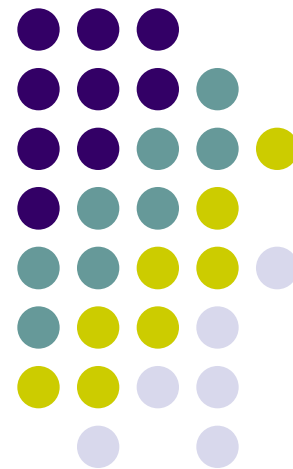


Циклические алгоритмы.



Задача 1. Вычислить сумму 6-ти первых натуральных чисел.



- **Входные данные:**

Промежуточная переменная:

i - натуральное число, числовой тип.

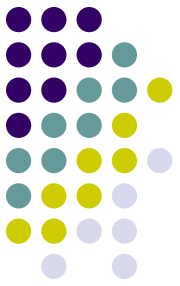
- **Выходные данные:**

S - сумма чисел, числовой тип.

$$S_0 = 0$$

$$i_0 = 0$$

$$\begin{array}{cccccccc} 1 & + & 2 & + & 3 & + & 4 & + & 5 & + & 6 \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ 0 + 1 = 1 & & 1 + 2 = 3 & & 3 + 3 = 6 & & 6 + 4 = 10 & & 10 + 5 = 15 & & 15 + 6 = 21 \end{array}$$



$$S_0=0 \quad 1 + 2 + 3 + 4 + 5 + 6$$

$$i_0=0 \quad S_0+i_1=S_1$$

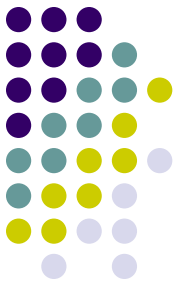
$$S_1+i_2=S_2$$

$$S_2+i_3=S_3$$

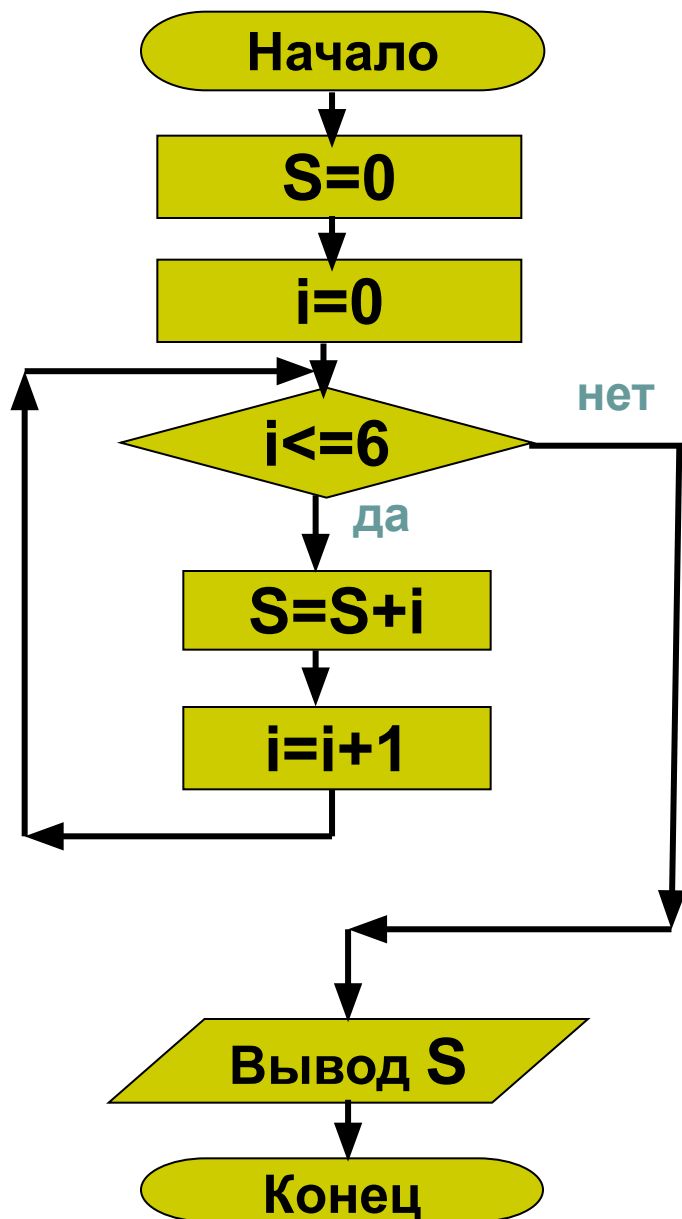
$$S_3+i_4=S_4$$

$$S_4+i_5=S_5$$

$$S_5+i_6=S_6$$



Блок-схема.



Трассировочная
таблица

Шаг цикла	S	i
	0	0
1	0	1
2	1	2
3	3	3
4	6	4
5	10	5
6	15	6
7	21	7
	21	

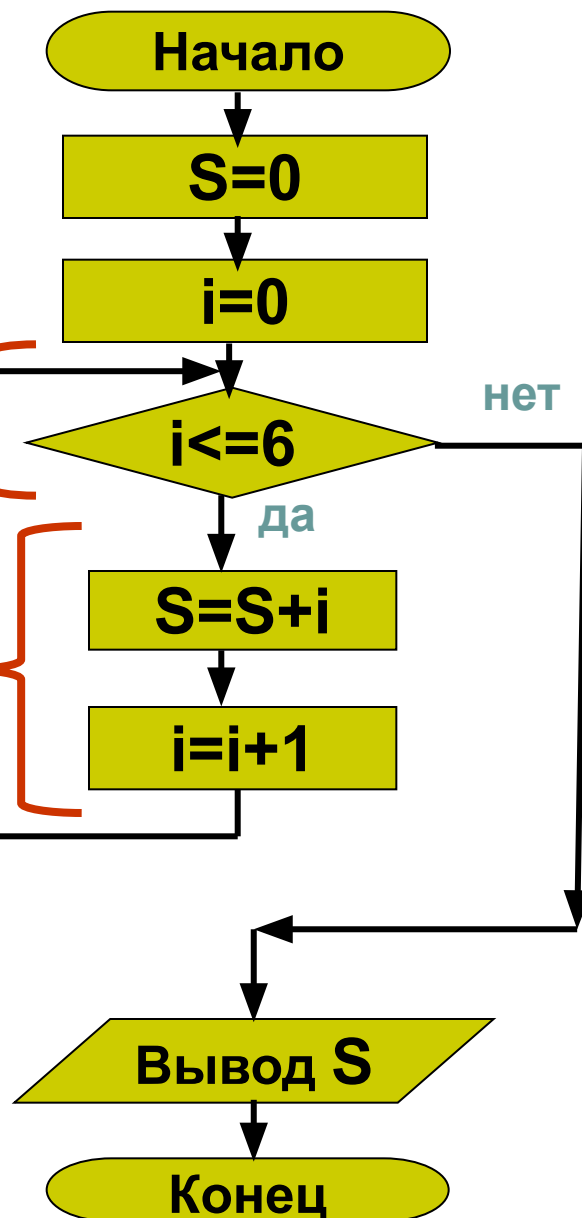
Блок-схема.



Циклический алгоритм

Условие цикла

Тело цикла



Циклический алгоритм содержит алгоритмическую структуру «**цикл**», позволяющую организовать повторение (выполнение более 2 раз) одной или нескольких команд.



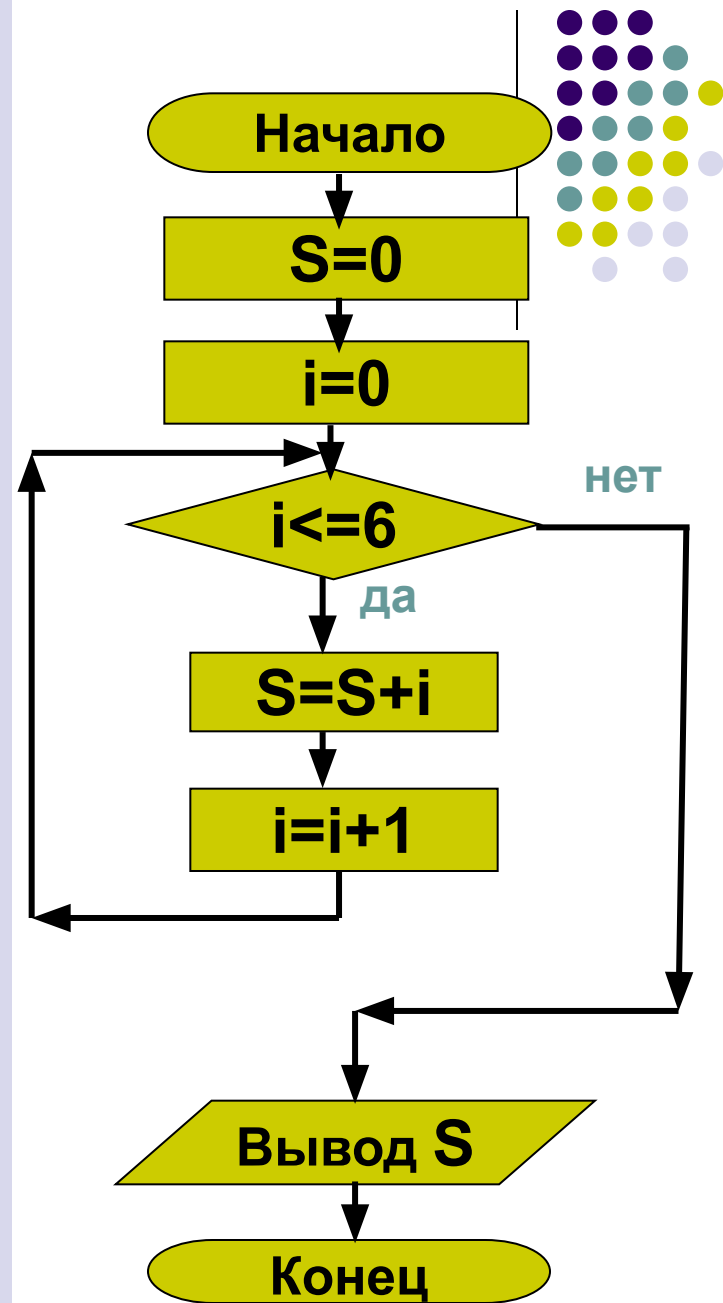
Оператор цикла

while <условие цикла> **do** <тело цикла>

Тело цикла выполняется до тех пор, пока условие принимает значение «истина».

Составим программу на языке Паскаль:

```
Program n1;  
uses crt;  
var i,S:integer;  
begin  
clrscr;  
S:=0;  
i:=0;  
while i<=6 do S:=S+i; i:=i+1;  
writeln('S=', S);  
end.
```



Задача 2. Вычислить сумму N первых натуральных чисел.



- **Входные данные:**

N – количество чисел, числовой тип

Промежуточная переменная:

i - натуральное число, числовой тип.

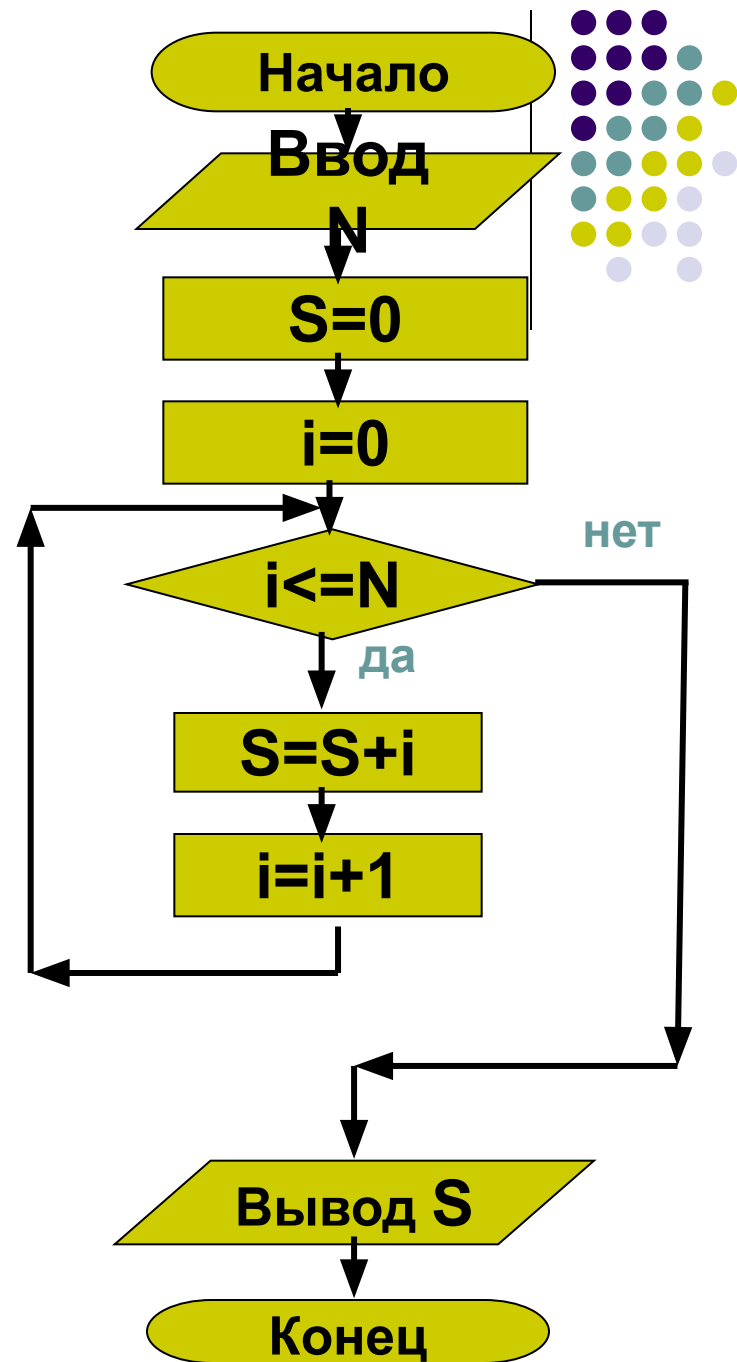
- **Выходные данные:**

S - сумма чисел, числовой тип.

Блок-схема.

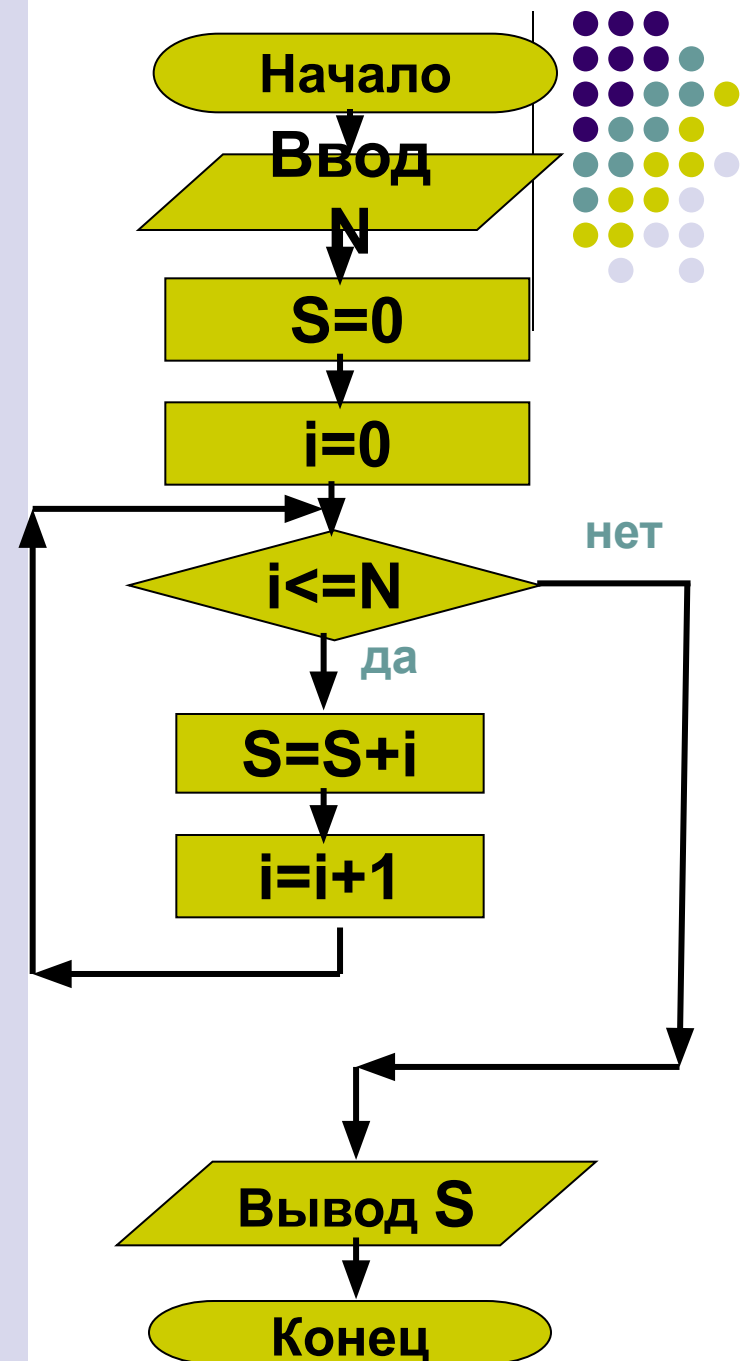
Трассировочная таблица

Шаг цикла	N	S	i
	5		
		0	0
1		0	1
2		1	2
3		3	3
4		6	4
5		10	5
6		15	6
		15	



Составим программу на языке Паскаль:

```
Program n2;  
uses crt;  
var N,i,S:integer;  
Begin  
clrscr;  
readln(N);  
S:=0;  
i:=0;  
while i<=N do S:=S+i; i:=i+1;  
writeln('S=', S);  
end.
```



Задача 3. Вычислить $n!$ (n -факториал)



$$1! = 1$$

$$2! = 1 \cdot 2 = 2$$

$$3! = 1 \cdot 2 \cdot 3 = 6$$

$$4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$$

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

...

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$$

Задача 3. Вычислить $n!$ (n -факториал)



- **Входные данные:**

N – количество чисел, числовой тип

Промежуточная переменная:

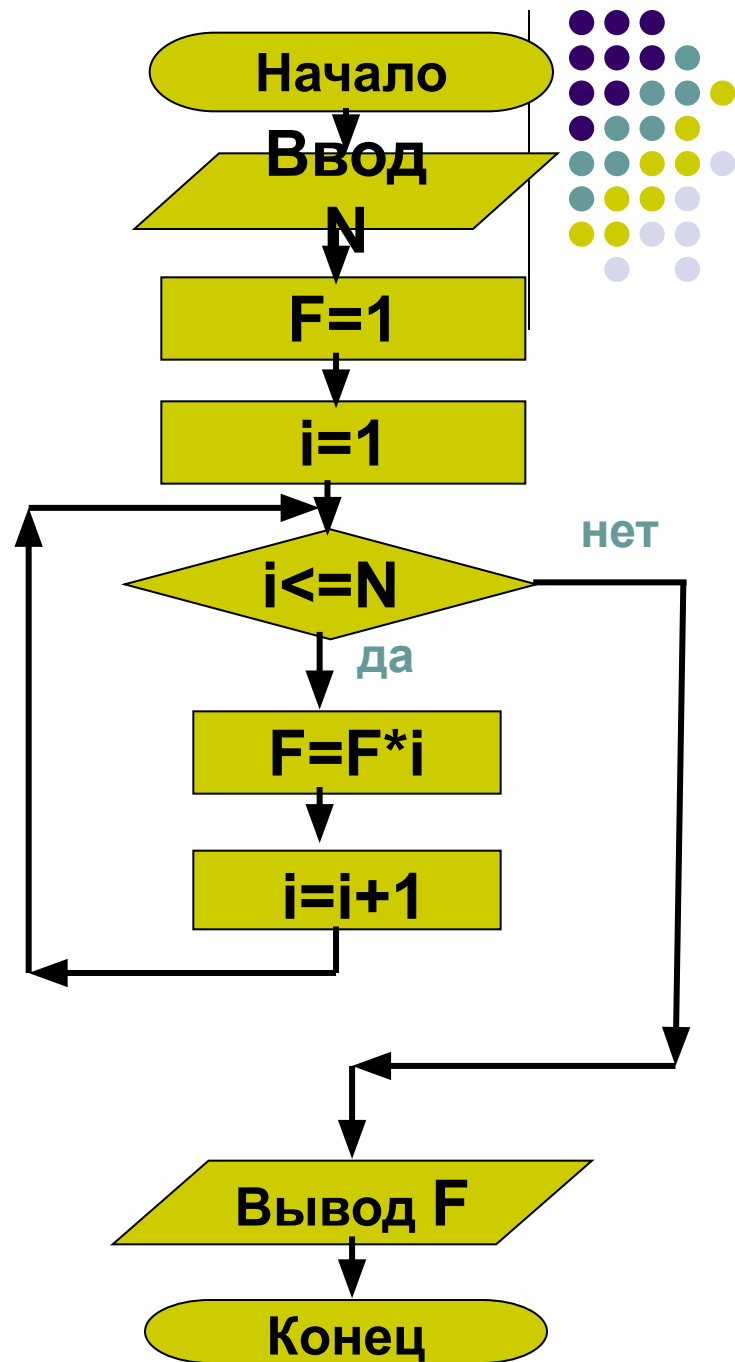
i - натуральное число, числовой тип.

- **Выходные данные:**

F - значение факториала, числовой тип.

Трассировочная таблица

Шаг цикла	N	F	i
	3		
		1	1
1		1	2
2		2	3
3		6	4
		6	



Составим программу на языке Паскаль:

Program n3;

uses crt;

var N,i,F:integer;

Begin

clrscr;

read(N);

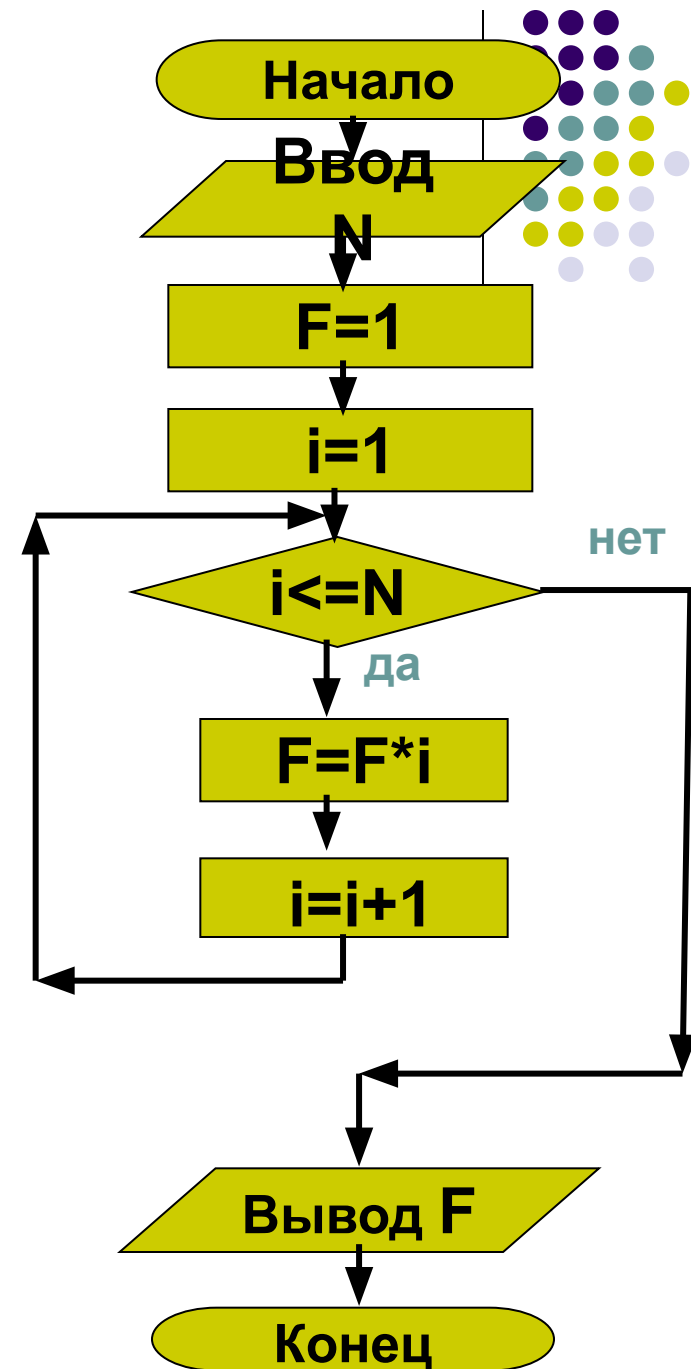
F:=1;

i:=1;

while i<=N do F:=F*i; i:=i+1;

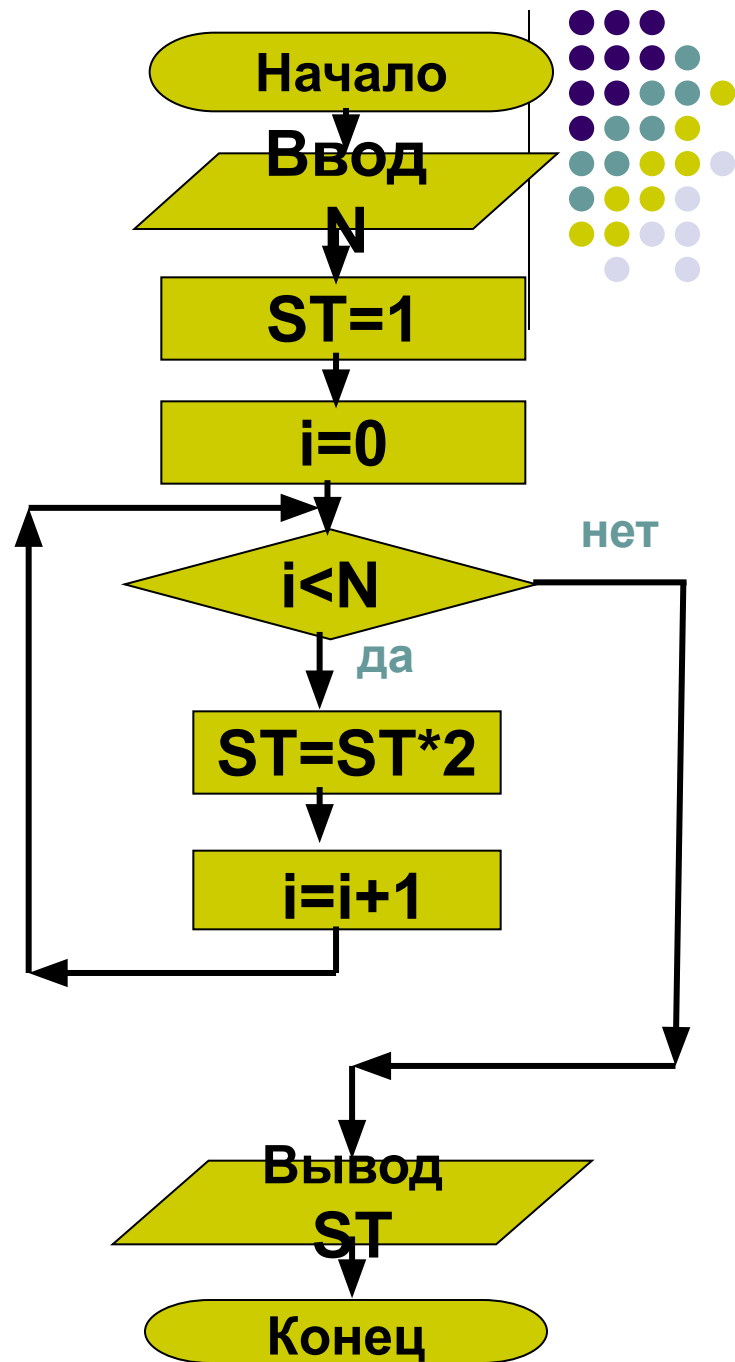
writeln('F=', F);

end.



Трассировочная таблица N

Шаг цикла	N	ST	i
		1	0
1		2	1
2		4	2
3		8	3
4		16	4
5		32	5
6		64	6
7		128	7



Задача 4. Вычислить 2^n



- **Входные данные:**

N – количество чисел, числовой тип

Промежуточная переменная:

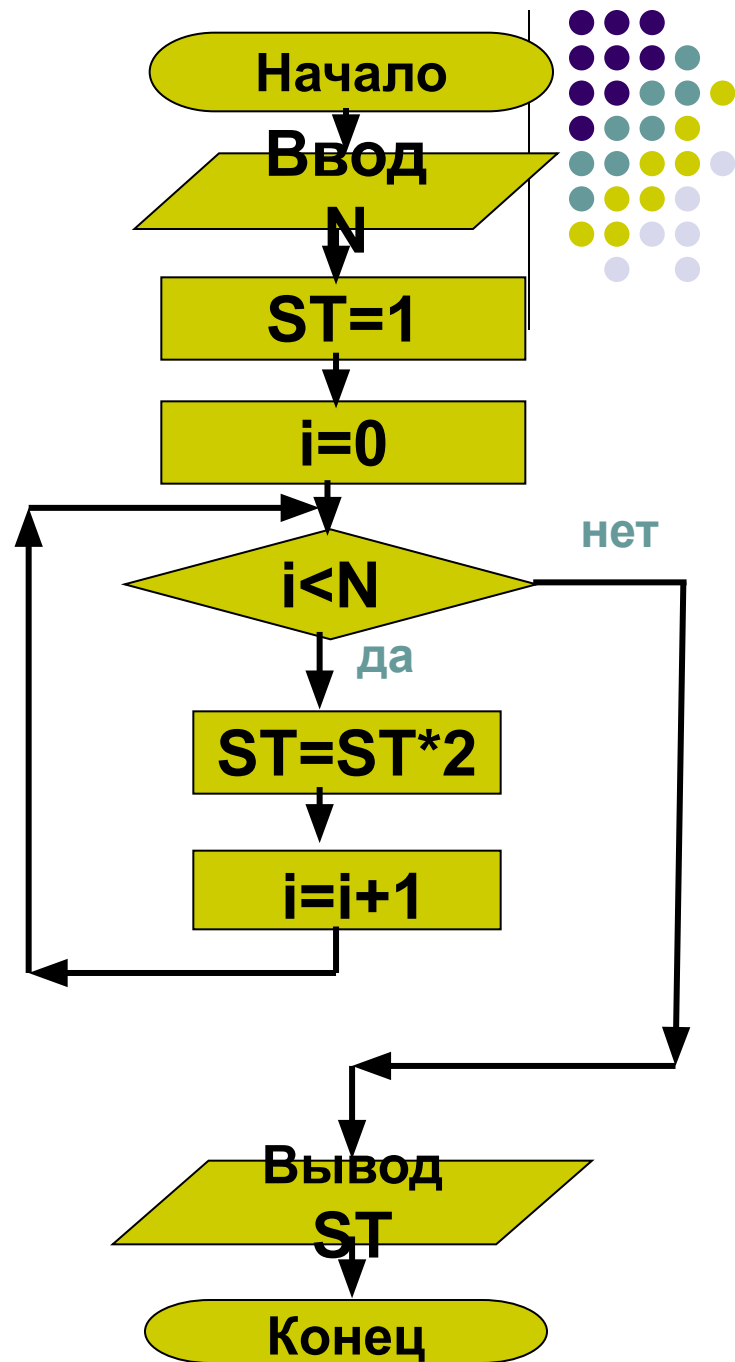
i - натуральное число, числовой тип.

- **Выходные данные:**

ST - значение степени, числовой тип.

Трассировочная таблица N

Шаг цикла	N	ST	i
		1	0
1		2	1
2		4	2
3		8	3
4		16	4
5		32	5
6		64	6
7		128	7



Составим программу на языке Паскаль:

Program n4;

```
uses crt;
```

```
var N,i,ST:integer;
```

```
Begin
```

```
clrscr;
```

```
read(N);
```

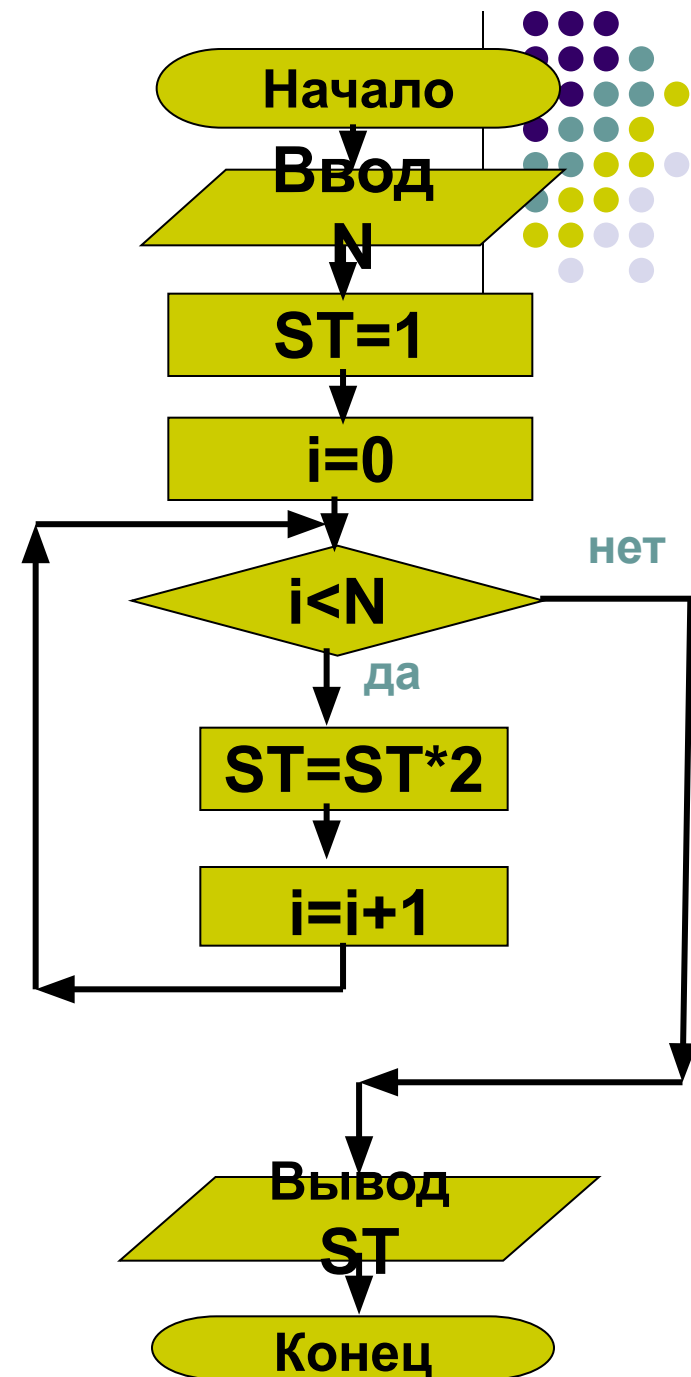
```
ST:=1;
```

```
i:=0;
```

```
while i<N do ST:=ST*2; i:=i+1;
```

```
writeln('ST=', ST);
```

```
end.
```

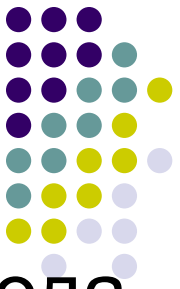


Виды циклических структур.



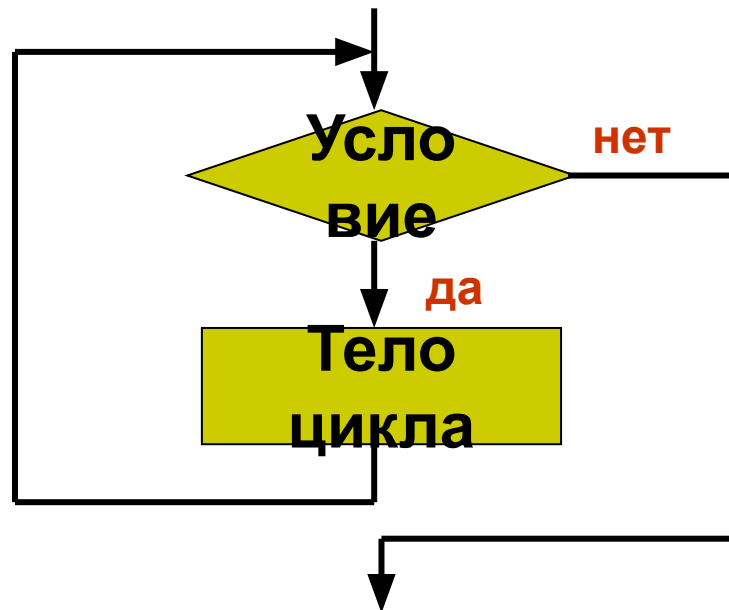
- Цикл с **предусловием**
- Цикл с **постусловием**
- Цикл с **условием окончания**

Цикл с предусловием.



- Проверка условия цикла находится до входа в цикл. Цикл может не выполниться ни разу.

while <условие цикла> **do** <тело цикла>



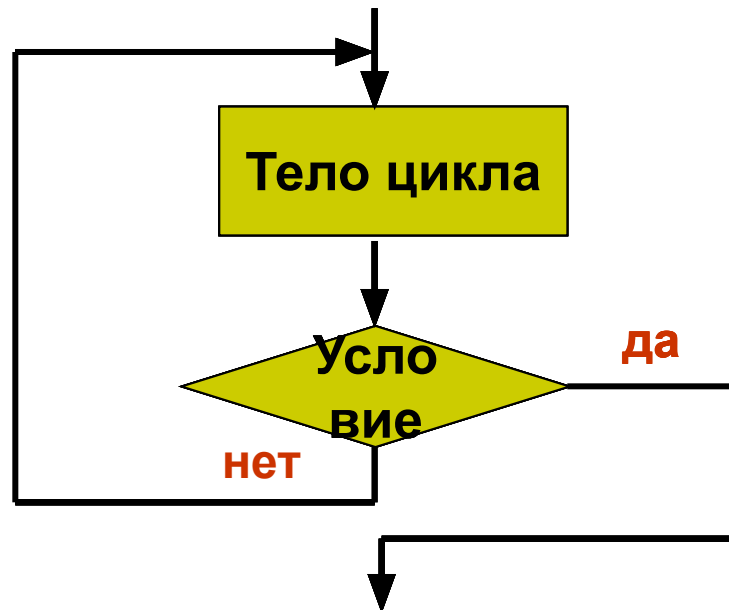
Цикл с постусловием.



- Проверка условия цикла находится после выполнения тела цикла.

Цикл выполнится хотя бы 1 раз.

repeat <тело цикла> **until** <условие цикла>





Цикл с условием окончания.

- Тело цикла выполняется определенное количество раз.

for $i=N$ **to** K **do** <тело цикла>

i -переменная (счетчик)

N -начальное значение

K -конечное значение

