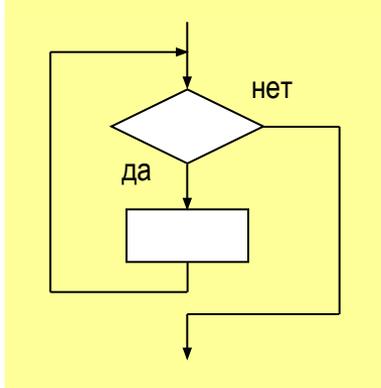


Циклические операторы на Паскале

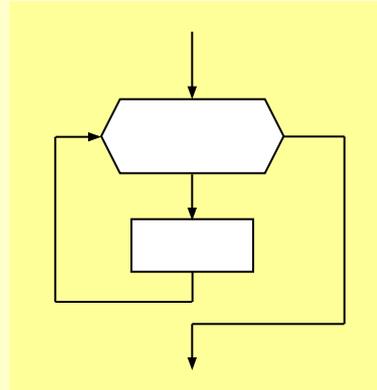


Оператор цикла с предусловием



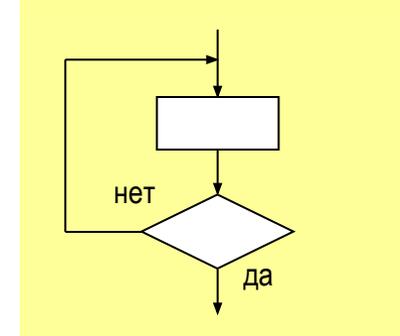
WHILE DO
 оператор

Оператор цикла с параметром



параметр цикла ↑	FOR	<input type="text" value="переменная"/>	:	<input type="text" value="выражение 1"/>	TO	<input type="text" value="выражение 2"/>	DO	<input type="text" value="оператор"/>
параметр цикла ↓	FOR	<input type="text" value="переменная"/>	:	<input type="text" value="выражение 1"/>	DOWNTO	<input type="text" value="выражение 2"/>	DO	<input type="text" value="оператор"/>
		<input type="text" value="Имя параметра (счетчика) цикла"/>		<input type="text" value="Начальное значение счетчика цикла"/>		<input type="text" value="Конечное значение счетчика цикла"/>		<input type="text" value="р"/>

Оператор цикла с постусловием



REPEAT
 оператор
 UNTIL



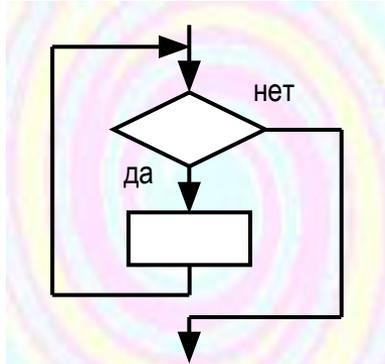
Цикл – это многократно выполняемая последовательность операторов (команд). Существует два основных цикла: цикл с предусловием и цикл с постусловием. В языке Паскаль используются три основные циклические конструкции: (WHILE ... DO...), (REPEAT ... UNTIL ...), (FOR ... TO / DOWNTO ... DO ...). Цикл с параметром (FOR ... DO ...) является частным случаем цикла с предусловием.



Циклические операторы на Паскале

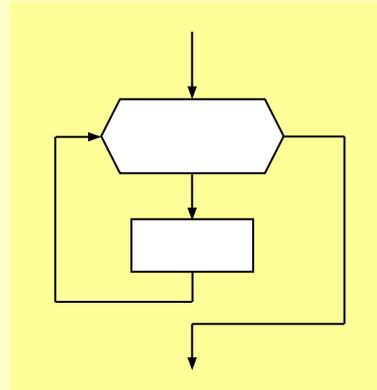


Оператор цикла с предусловием



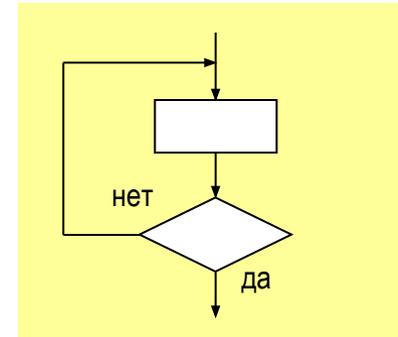
WHILE *услови* DO
 оператор *p*

Оператор цикла с параметром



параметр цикла ↑	FOR	переменна	:	выражение 1	TO	выражение 2	DO	оператор
	R	я	=		O		O	p
параметр цикла ↓	FOR	переменна	:	выражение 1	DOWNT	выражение 2	DO	оператор
	R	я	=		O		O	p
Имя параметра (счетчика) цикла				Начальное значение счетчика цикла		Конечное значение счетчика цикла		

Оператор цикла с постусловием



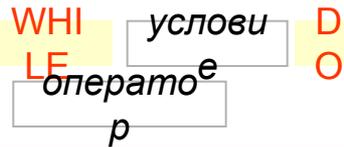
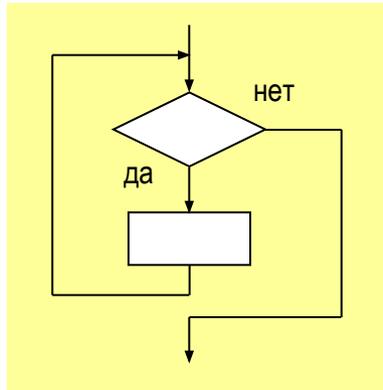
REPEAT
 оператор
 UNTIL *услови*

Цикл с предусловием (WHILE ... DO ...) многократно выполняет одни и те же действия при истинности условия, которое изменяется обычно внутри цикла. Истинность условия проверяется перед выполнением операторов. Таким образом, если условие с самого начала оказалось ложным, то операторы тела цикла не будут выполнены ни разу. Если в цикле необходимо выполнить несколько простых операторов, они объединяются после служебного слова DO в составной оператор операторными скобками BEGIN ... END.

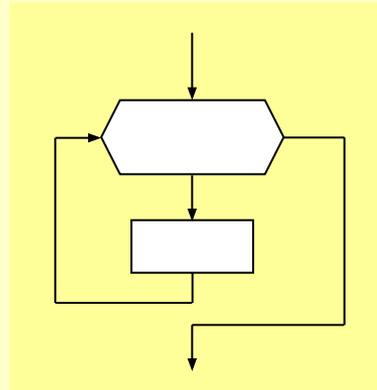
Циклические операторы на Паскале



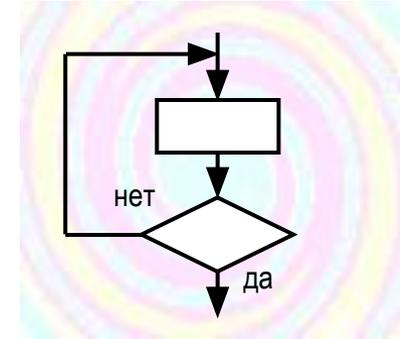
Оператор цикла с предусловием



Оператор цикла с параметром



Оператор цикла с постусловием

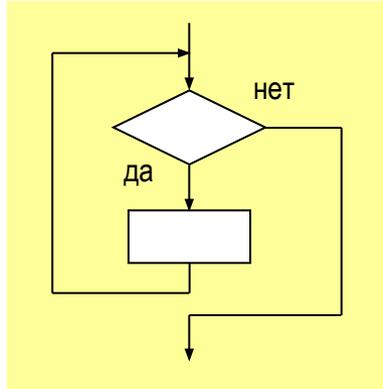


В цикле с постусловием (REPEAT...UNTIL...) сначала выполняются операторы, составляющие тело цикла, затем проверяется условие, которое в данном случае является условием выхода из цикла, т. е. если оно ложно, то операторы цикла повторяются, иначе (если условие истинно) – цикл завершается. Таким образом, в конструкции (REPEAT...UNTIL...) тело цикла всегда выполняется хотя бы один раз. В данном цикле не требуется обязательное использование операторных скобок при циклическом выполнении нескольких операторов.

Циклические операторы на Паскале

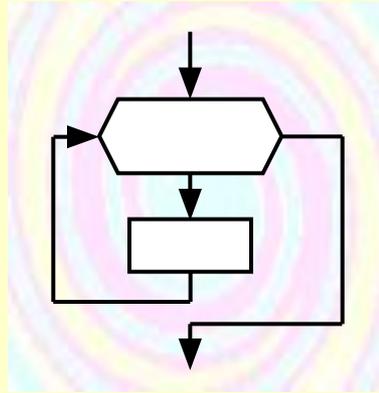


Оператор цикла с предусловием

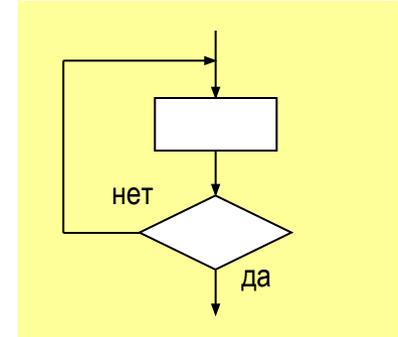


WHILE **услови** DO
оператор

Оператор цикла с параметром



Оператор цикла с постусловием



REPEAT
оператор
 UNTIL **услови**

параметр цикла ↑	FOR	переменная	:	выражение 1	TO	выражение 2	DO	оператор
параметр цикла ↓	FOR	переменная	:	выражение 1	DOWNTO	выражение 2	DO	оператор
		я	=					р
		я	=					р
		Имя параметра (счетчика) цикла		Начальное значение счетчика цикла		Конечное значение счетчика цикла		

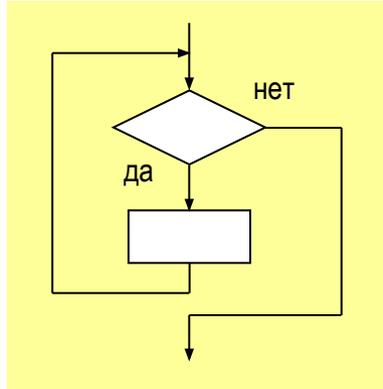
Цикл (FOR ... TO ... DO ...) работает следующим образом: параметру цикла присваивается значение выражения 1, потом это значение сравнивается со значением выражения 2. Если условие выполнения цикла истинно (значение счетчика меньше значения выражения 2), то выполняется тело цикла. Далее значение параметра 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.



Циклические операторы на Паскале

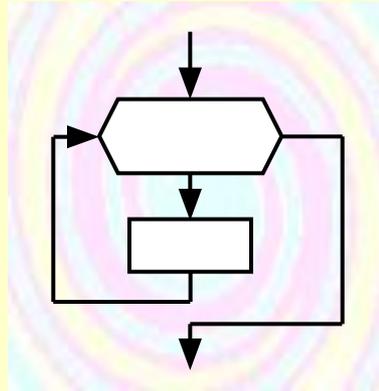


Оператор цикла с предусловием

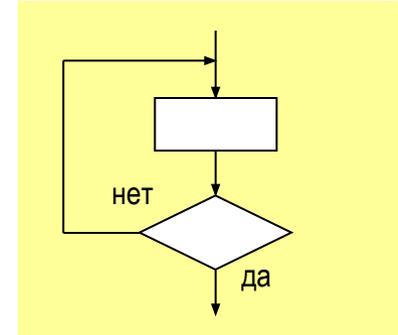


WHILE *услови* DO
LE *оператор* E
P

Оператор цикла с параметром



Оператор цикла с постусловием



REPEAT
T *оператор*
UNTIL *услови* E
P

параметр цикла ↑	FOR	переменная	:	выражение 1	TO	выражение 2	DO	оператор
параметр цикла ↓	FOR	переменная	:	выражение 1	DOWNTO	выражение 2	DO	оператор
		я	=					р
		я	=					р
		Имя параметра (счетчика) цикла		Начальное значение счетчика цикла		Конечное значение счетчика цикла		

Цикл (FOR ... DOWNTO ... 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 больше значения выражения 2), то выполняется тело цикла. Далее значение параметра цикла автоматически уменьшается на единицу, снова сравнивается со значением выражения 2, если условие истинно, то опять выполняется тело цикла. Цикл прекращается, когда значение параметра цикла станет меньше значения выражения 2.

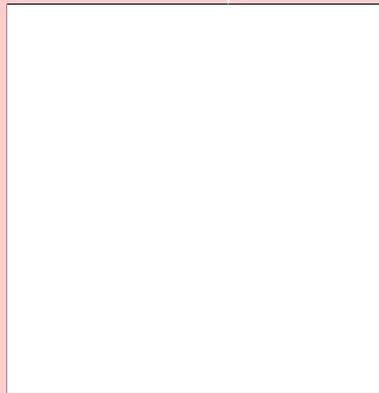
Циклические операторы на Паскале



```
PROGRAM FAKTORIA ;
VAR M, n, r, F : INTEGER ;
BEGIN
  N := READLN ( n ) ;
  F := 1 ;
```

«N факториал» $F = N! = 1 \cdot 2 \cdot \dots \cdot N$

```
  r := 1 ;
  WHILE r <= n DO
  BEGIN
    F := F * r ;
    r := r + 1 ;
  END ;
```

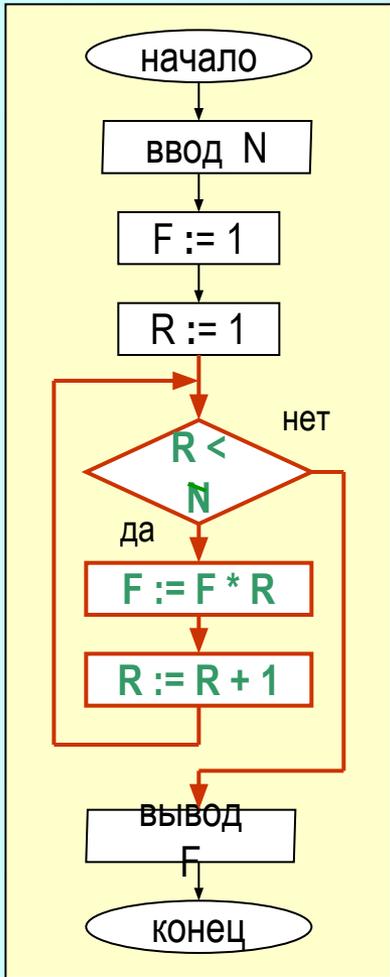


```
  WRITELN ( F ) ;
END .
```

Для решения одной и той же задачи можно составить несколько вариантов вычислительных алгоритмов на основе применения различных операторов. При этом, чем сложнее задача, тем больше возможных путей ее решения. На данном слайде предлагается четыре варианта программ на языке Паскаль для вычисления N факториала. Это хороший пример для демонстрации всех циклических конструкций, используемых в Паскале.



Циклические структуры алгоритмов



«N факториал»
 $F = N! = 1 * 2 * \dots * N$

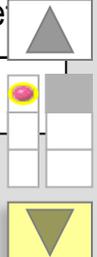
```

алг N!
цел F, N, R
нач ввод N
  F := 1
  R := 1
  пока R <= N,
    повторять
      F := F * R
      R := R + 1
  нц
  вывод F
кц
кон
  
```

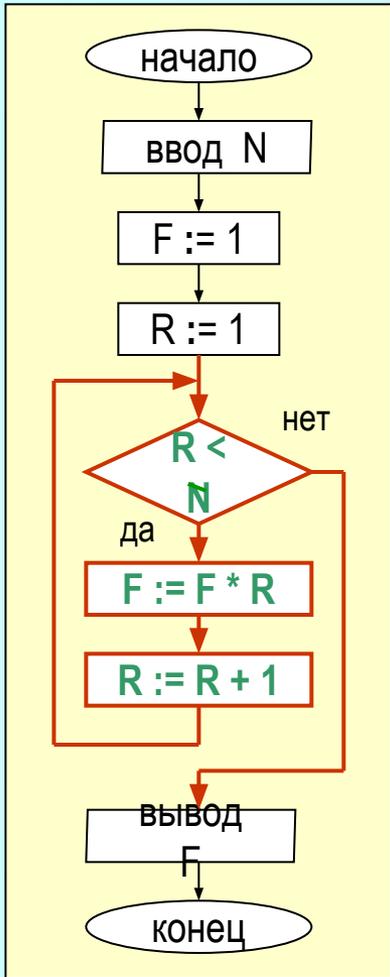
№	Операция	N	F	R	Условие
1	Ввод N	3			
2	F := 1		1		
3	R := 1			1	
4	R <= N				1 <= 3 да
5	F := F * R		1		
6	R := R + 1			2	
7	R <= N				2 <= 3 да
8	F := F * R		2		
9	R := R + 1			3	
10	R <= N				3 <= 3 да
11	F := F * R		6		
12	R := R + 1			4	
13	R <= N				4 <= 3 нет
14	Вывод		6		

Количество различных комбинаций из N предметов, получаемых изменением их порядка, называется числом перестановок. Это число выражается функцией от N, которая называется факториалом (N!) $F = N! = 1 * 2 * \dots * N$ Подобные задачи решает раздел математики - КОМБИНАТОРИКА.

Для решения подобных задач для разных значений N необходимо выполнить различное число умножений, поэтому алгоритм решения данной задачи будет циклическим.



Циклические структуры алгоритмов



«N факториал»
 $F = N! = 1 * 2 * \dots * N$

```

алг N!
цел F, N, R
нач ВВОД N
  F := 1
  R := 1
  пока R <= N,
    повторять
      F := F * R
      R := R + 1
  кц
  вывод F
кон
  
```

№	Операция	N	F	R	Условие
1	Ввод N	3			
2	F := 1		1		
3	R := 1			1	
4	R <= N				1 <= 3 да
5	F := F * R		1		
6	R := R + 1			2	
7	R <= N				2 <= 3 да
8	F := F * R		2		
9	R := R + 1			3	
10	R <= N				3 <= 3 да
11	F := F * R		6		
12	R := R + 1			4	
13	R <= N				4 <= 3 не
14	Вывод F		6		

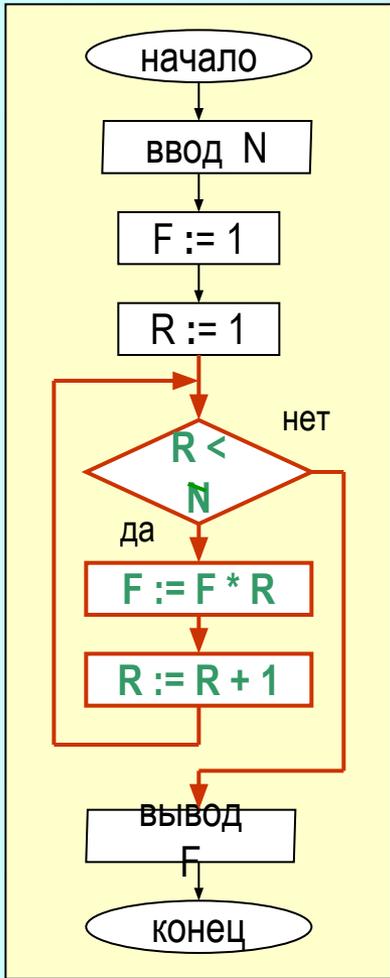
ЦИКЛ – это команда исполнителю многократно повторить указанную последовательность команд.

В нашем примере применяется алгоритмическая структура «цикл с предусловием»

Пока выполняется условие цикла ($R \leq N$),
 повторяется выполнение тела цикла ($F := F * R, R := R + 1$)



Циклические структуры алгоритмов



«N факториал»
 $F = N! = 1 * 2 * \dots * N$

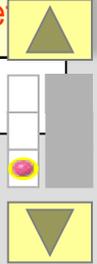
```

алг N!
цел F, N, R
нач ввод N
  F := 1
  R := 1
  пока R <= N,
    повторять
  нц
    F := F * R
    R := R + 1
  кц
  вывод F
кон
  
```

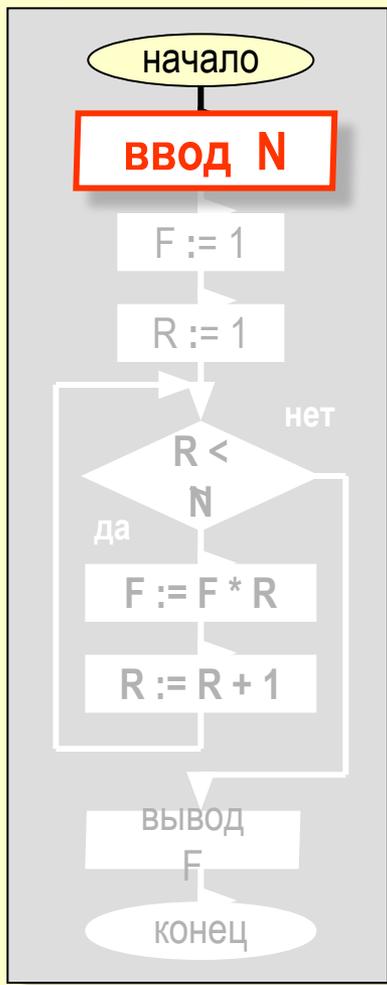
№	Операция	N	F	R	Условие
1	Ввод N	3			
2	F := 1		1		
3	R := 1			1	
4	R <= N				1 <= 3 да
5	F := F * R		1		
6	R := R + 1			2	
7	R <= N				2 <= 3 да
8	F := F * R		2		
9	R := R + 1			3	
10	R <= N				3 <= 3 да
11	F := F * R		6		
12	R := R + 1			4	
13	R <= N				4 <= 3 не
14	вывод		6		

В данном алгоритме переменная R - множитель, значение которого меняется от 1 до N через 1. Произведение накапливается в переменной F, начальное значение которой равно 1.

Цикл заканчивается, когда R = N + 1

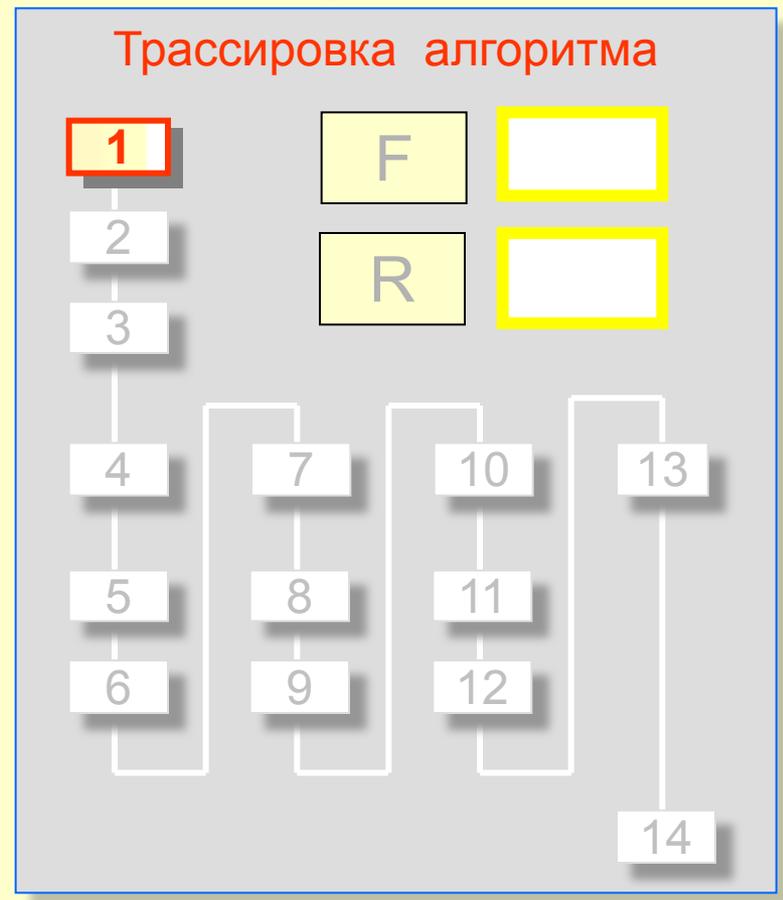


Трассировка циклического алгоритма



```

алг N!
цел F, N, R
нач ВВОД N      N = 3
    F := 1
    R := 1
    пока R <= N, повторять
        нц
            F := F * R
            R := R + 1
        кц
    вывод F
кон
  
```

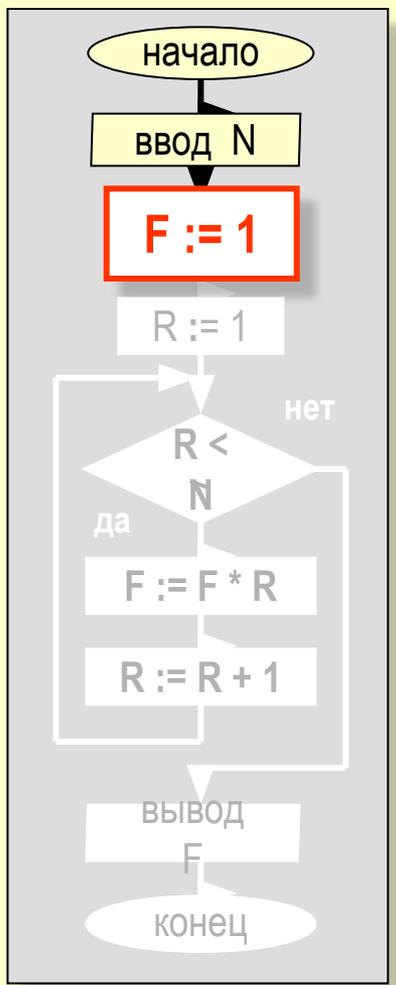


№	Операция	N	F	R	Условие
1	Ввод N	3			

На первом шаге алгоритма необходимо ввести исходные данные для решения задачи. В нашем примере для вычисления факториала необходимо задать количество (N) предметов. *Рассмотрим случай, когда N = 3.*

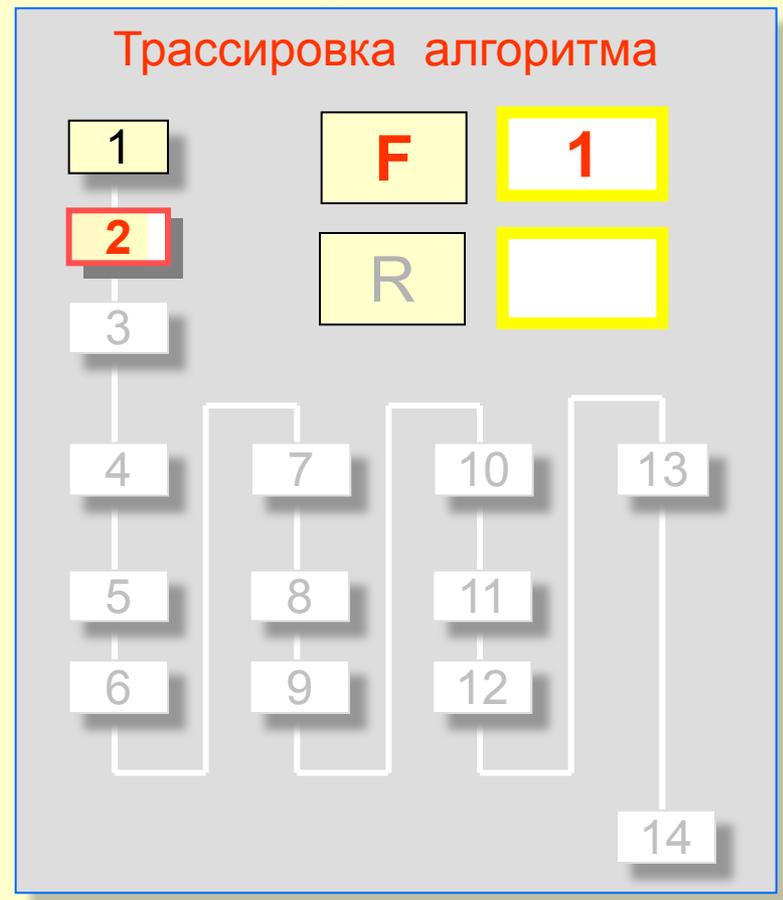


Трассировка циклического алгоритма



```

алг N!
цел F, N, R
нач ВВОД N      N = 3
                F := 1
                R := 1
                пока R <= N, повторять
                нц
                    F := F * R
                    R := R + 1
                кц
                ВЫВОД F
конец
  
```

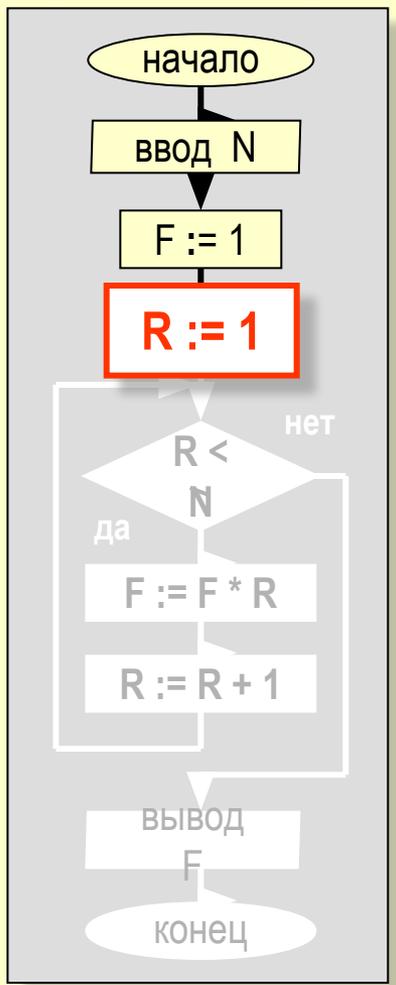


№	Операция	N	F	R	Условие
2	F := 1		1		

На втором шаге алгоритма функции F присваивается начальное значение, равное единице (F := 1)

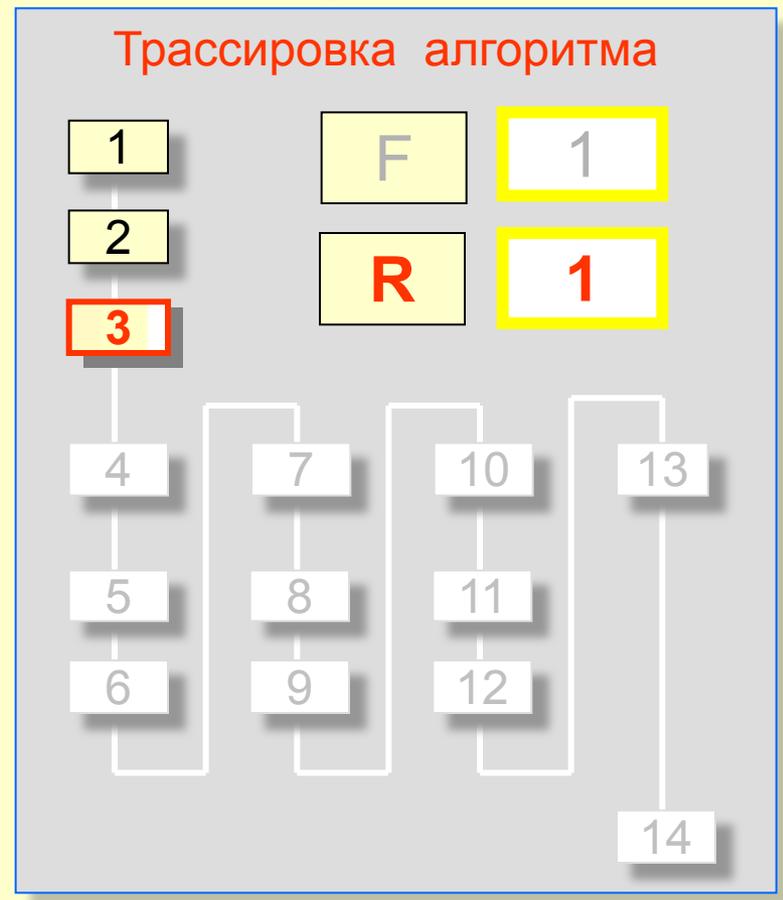


Трассировка циклического алгоритма



```

алг N!
цел F, N, R
нач  ВВОД N      N = 3
      F := 1
      R := 1
      пока R <= N, повторять
      НЦ
          F := F * R
          R := R + 1
      КЦ
      ВЫВОД F
конец
  
```

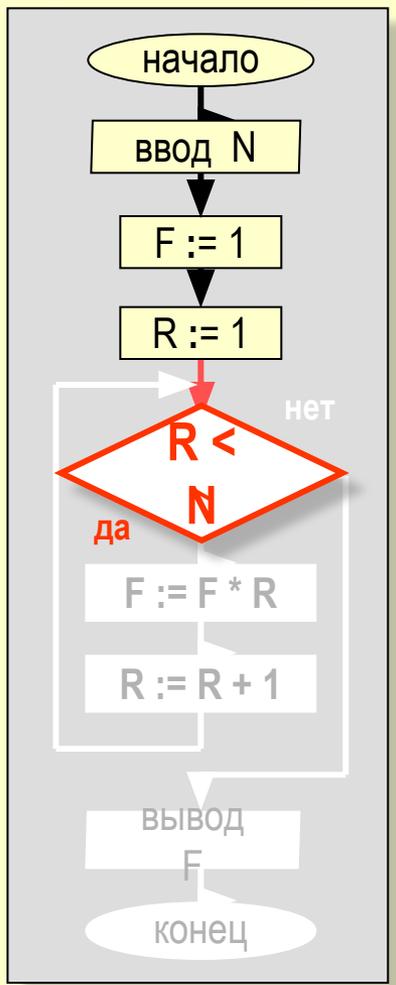


№	Операция	N	F	R	Условие
3	R := 1			1	

Промежуточной переменной R (счетчику) присваивается начальное значение, равное 1. Эта переменная в алгоритме вычисления факториала играет роль множителя, значение которого меняется от начального значения до N (в нашем примере до 3)

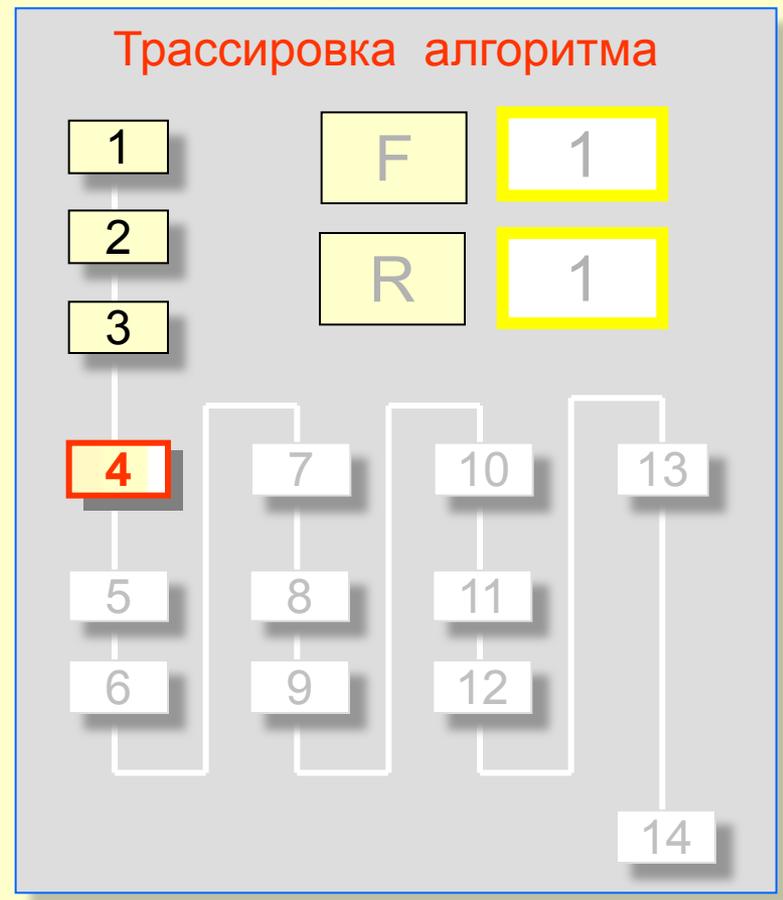


Трассировка циклического алгоритма



```

алг N!
цел F, N, R
нач ввод N
    N = 3
    F := 1
    R := 1
    пока R <= N, повторять
        F := F * R
        R := R + 1
    кц
    вывод F
кон
  
```

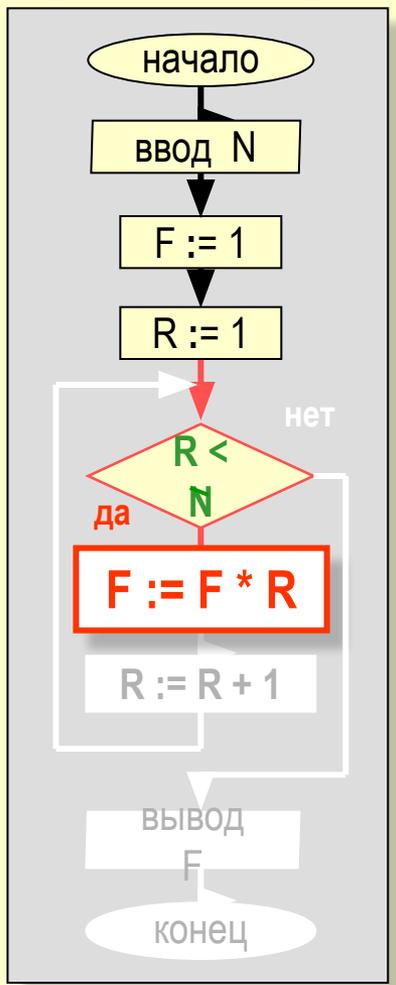


№	Операция	N	F	R	Условие
4	R <= N				1 <= 3 да

Проверяется условие выполнения цикла $R \leq N$. От истинности этого условия зависит, будут ли исполняться команды, составляющие тело цикла. Пока условие на этом шаге **выполняется**, так как $R=1$, а $N=3$ ($1 < 3$)

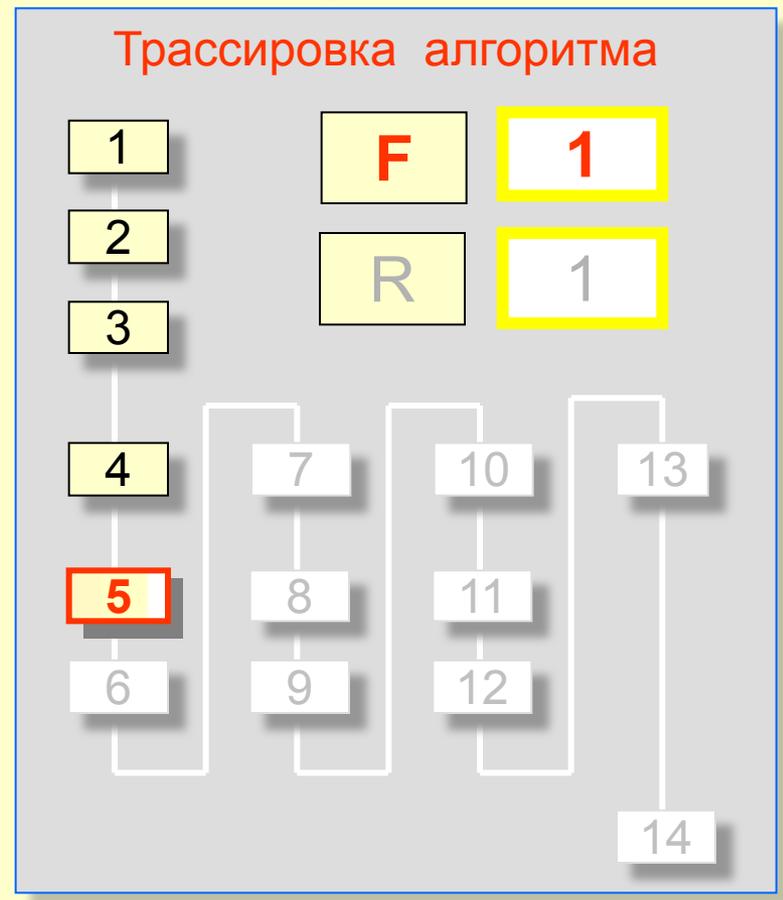


Трассировка циклического алгоритма



```

алг N!
цел F, N, R
нач ввод N
    N = 3
    F := 1
    R := 1
    пока R <= N, повторять
        F := F * R
        R := R + 1
    кц
    вывод F
кон
  
```

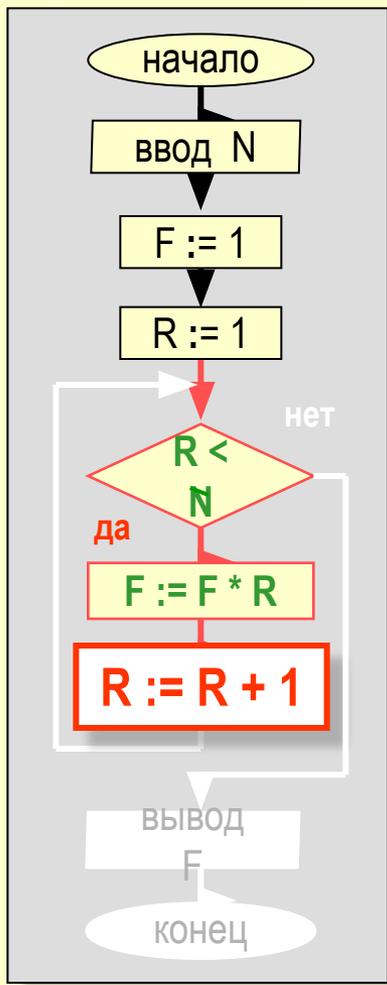


№	Операция	N	F	R	Условие
5	F := F * R		1		

Поскольку условие выполнения цикла на шаге №4 выполнено, исполняются последовательно две команды вычисления факториала, составляющие тело цикла. На этом шаге функции F присваивается значение 1 – результат перемножения предыдущих значений переменных F и R ($F := F * R = 1 * 1 = 1$)

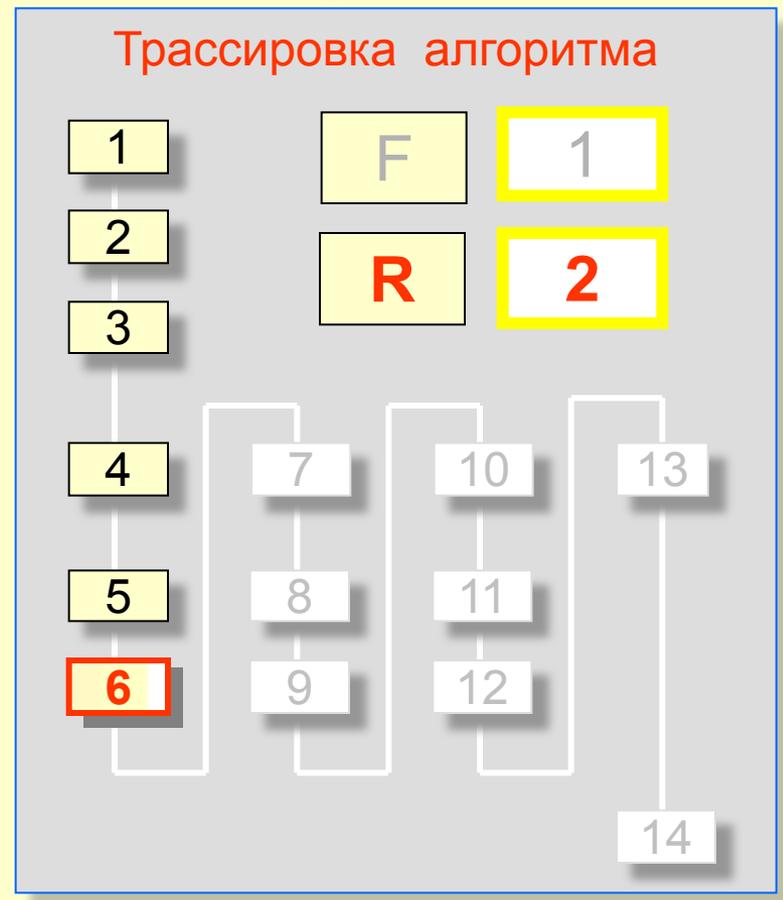


Трассировка циклического алгоритма



```

алг N!
цел F, N, R
нач  ВВОД N      N = 3
      F := 1
      R := 1
пока R <= N, повторять
НЦ
      F := F * R
      R := R + 1
кц
ВЫВОД F
конец
    
```

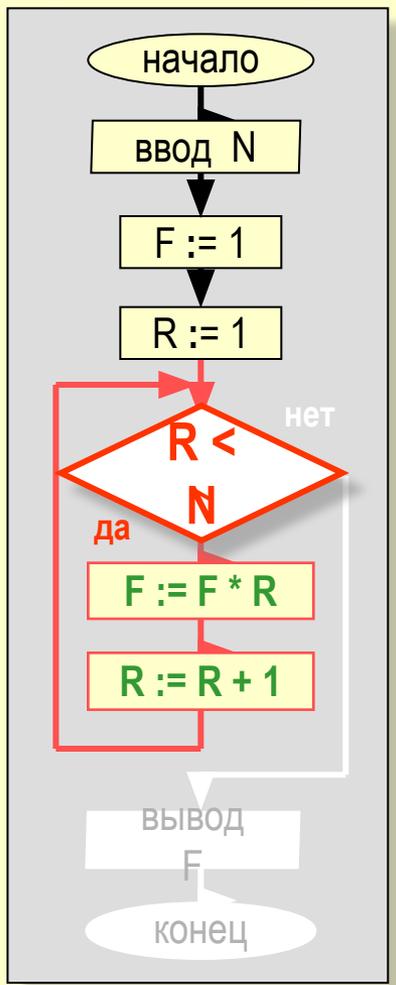


№	Операция	N	F	R	Условие
6	R := R + 1			2	

Значение переменной R увеличивается на единицу ($R := R + 1 = 1 + 1 = 2$).
 Новое значение множителя (переменной R) равно 2.

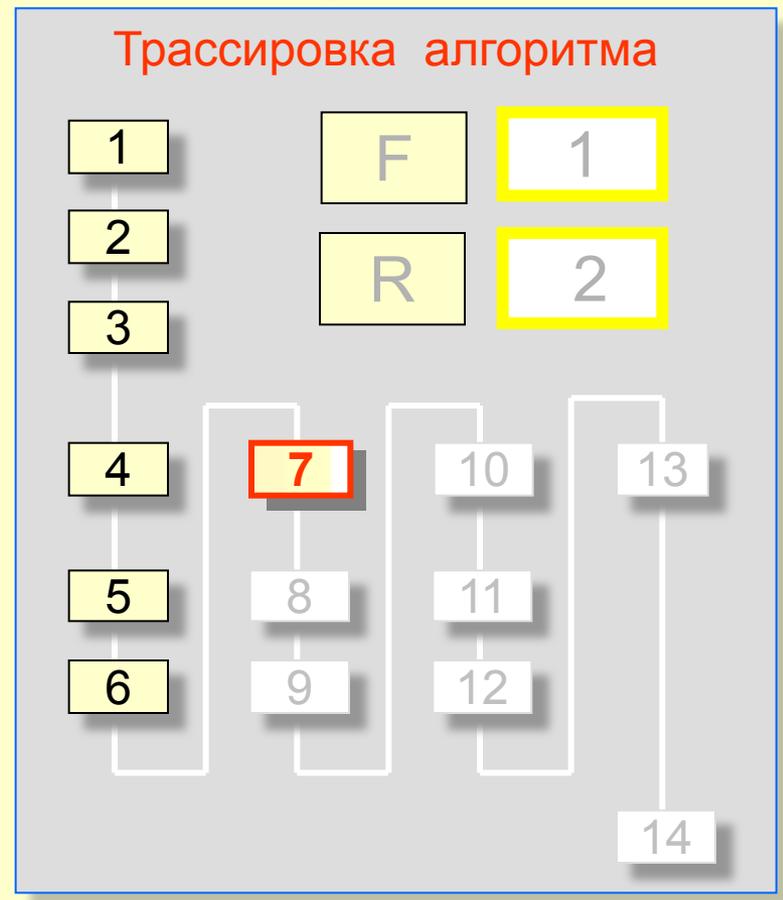


Трассировка циклического алгоритма



```

алг N!
цел F, N, R
нач ввод N
    N = 3
    F := 1
    R := 1
    пока R <= N, повторять
        F := F * R
        R := R + 1
    кц
    вывод F
конец
  
```

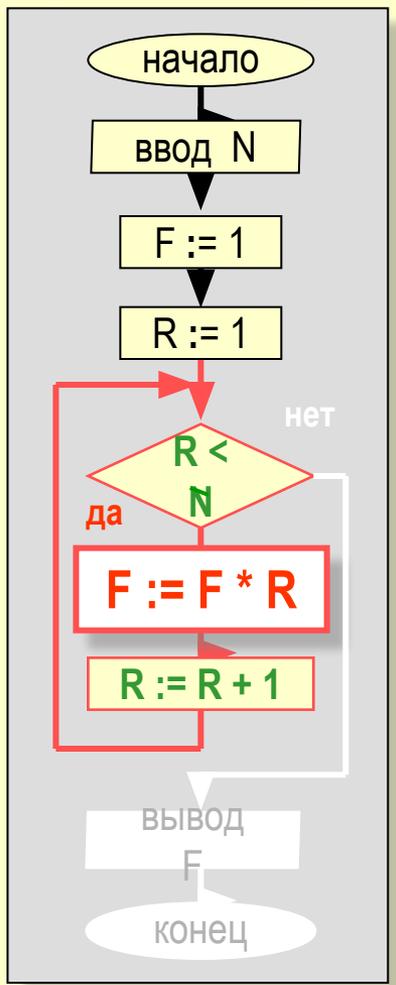


№	Операция	N	F	R	Условие
7	R <= N				2 <= 3 да

Второй раз проверяется условие выполнения цикла $R \leq N$ при новом значении R . Пока условие на этом шаге **выполняется**, так как $R = 2$, а $N = 3$ ($2 < 3$)



Трассировка циклического алгоритма



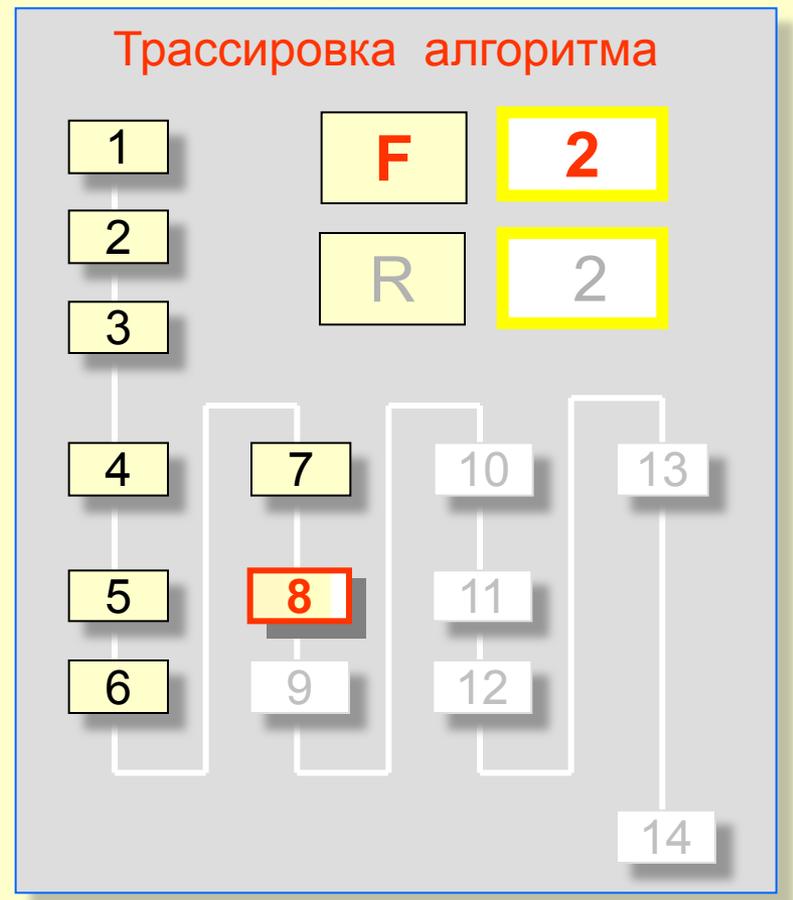
```

алг N!
цел F, N, R
нач  ВВОД N      N = 3

      F := 1
      R := 1

пока R <= N, повторять
    F := F * R
    R := R + 1
кц

ВЫВОД F
конец
    
```

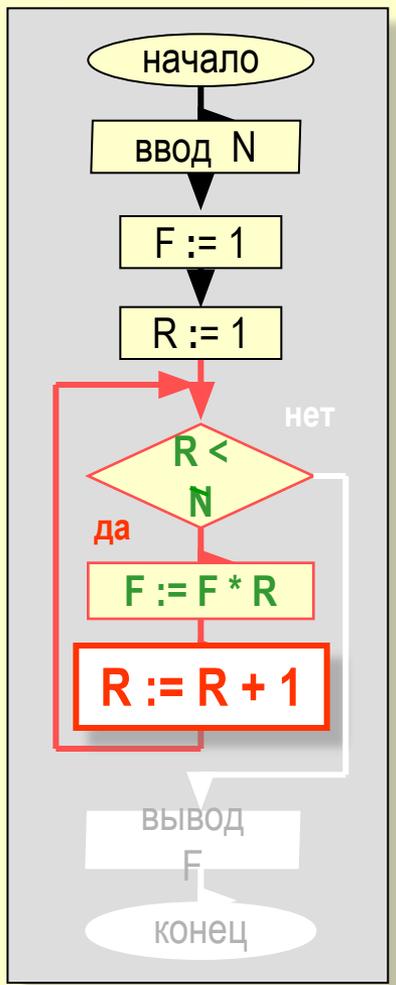


№	Операция	N	F	R	Условие
8	F := F * R		2		

Поскольку условие выполнения цикла на шаге № 7 выполнено, во второй раз исполняются последовательно две команды вычисления факториала, составляющие тело цикла. На этом шаге функции F присваивается значение 2 – результат перемножения предыдущих значений переменных F и R ($F := F * R = 1 * 2 = 2$)



Трассировка циклического алгоритма

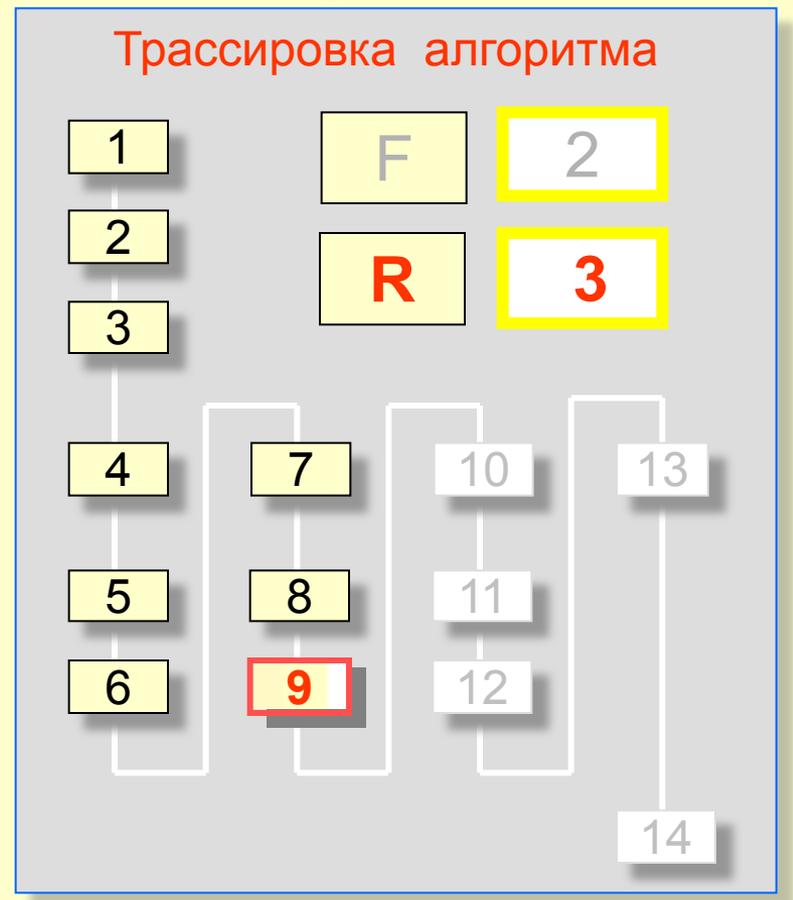


```

алг N!
цел F, N, R
нач  ВВОД N      N = 3

      F := 1
      R := 1

пока R <= N, повторять
НЦ
    F := F * R
    R := R + 1
кц
ВЫВОД F
кОН
    
```

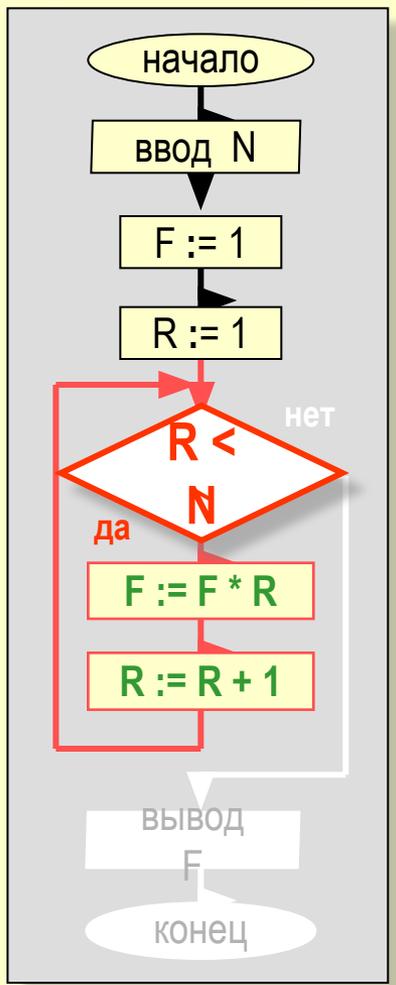


№	Операция	N	F	R	Условие
9	R := R + 1			3	

Значение переменной R увеличивается на единицу ($R := R + 1 = 2 + 1 = 3$).
 Новое значение множителя (переменной R) равно 3.

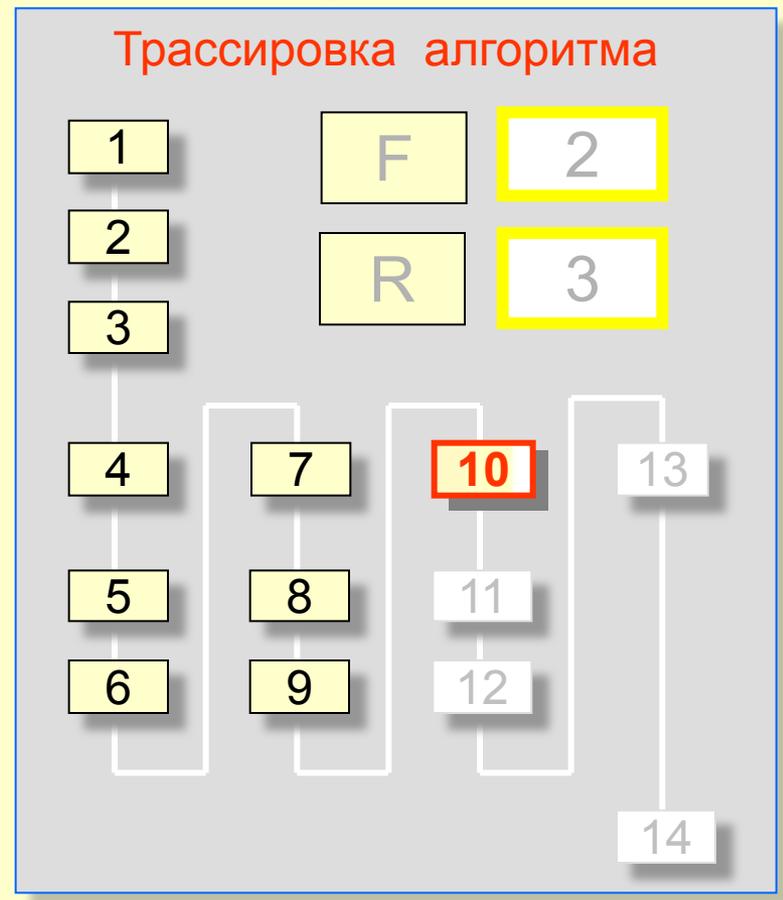


Трассировка циклического алгоритма



```

алг N!
цел F, N, R
нач ввод N
    N = 3
    F := 1
    R := 1
    пока R <= N, повторять
        F := F * R
        R := R + 1
    кц
    вывод F
кон
  
```

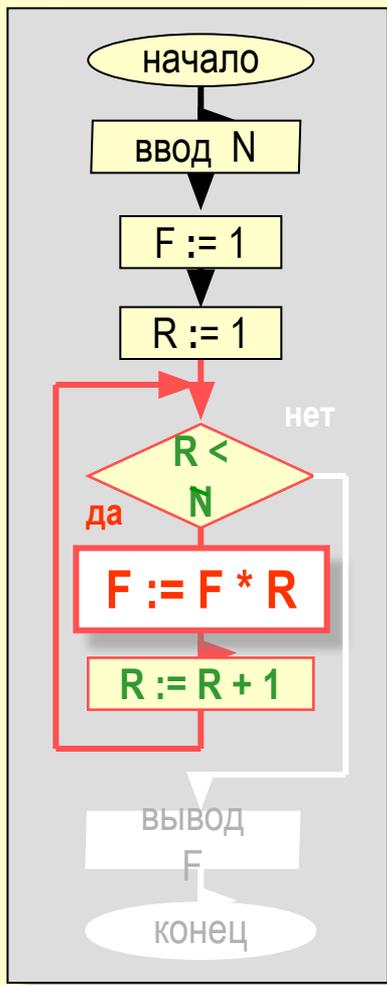


№	Операция	N	F	R	Условие
10	R <= N				3 <= 3 да

В третий раз проверяется условие выполнения цикла $R \leq N$ при новом значении R. Пока условие на этом шаге **выполняется**, так как $R = 3$, а $N = 3$ ($3 = 3$)



Трассировка циклического алгоритма

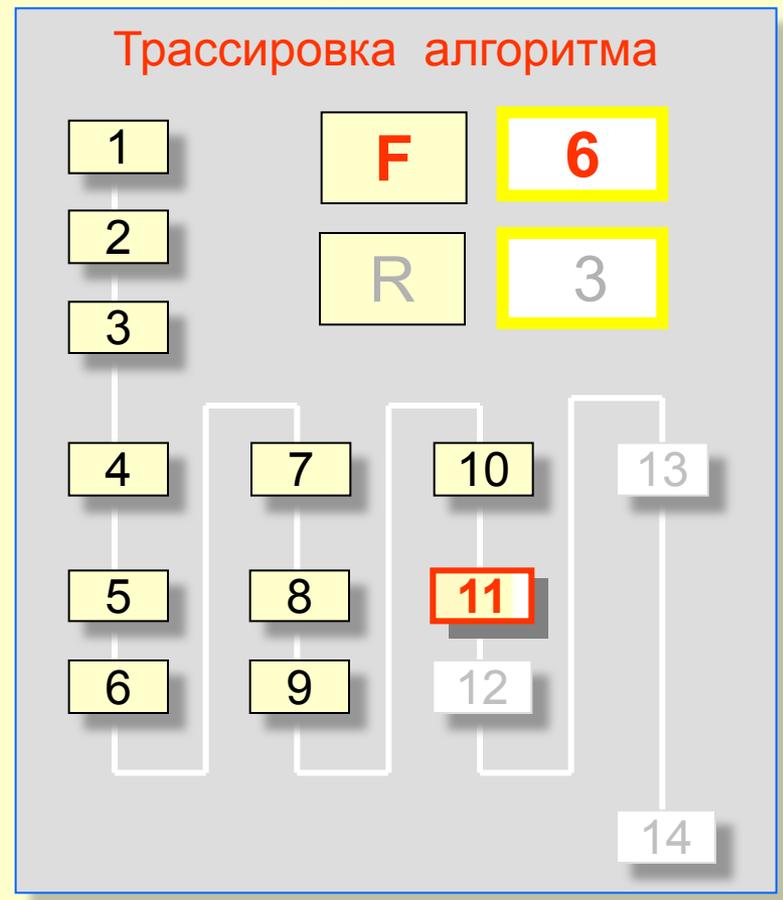


```

алг N!
цел F, N, R
нач  ВВОД N      N = 3

      F := 1
      R := 1

пока R <= N, повторять
    нц
        F := F * R
        R := R + 1
    кц
ВЫВОД F
кОН
    
```

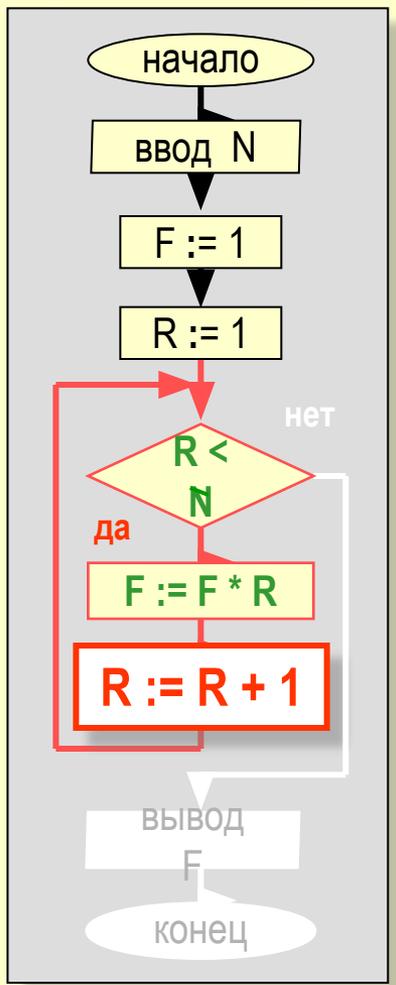


№	Операция	N	F	R	Условие
11	F := F * R		6		

Поскольку условие выполнения цикла на шаге № 10 выполнено, в третий раз исполняются последовательно две команды вычисления факториала, составляющие тело цикла. На этом шаге функции F присваивается значение 6 – результат перемножения предыдущих значений переменных F и R ($F := F * R = 2 * 3 = 6$)



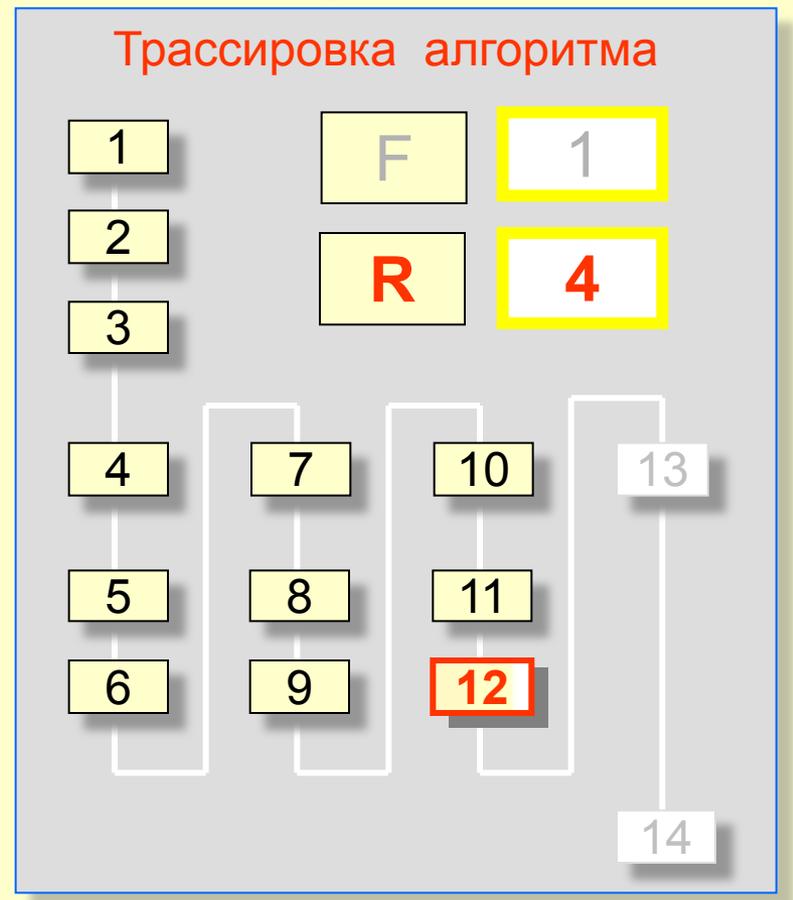
Трассировка циклического алгоритма



```

алг N!
цел F, N, R
нач  ВВОД N      N = 3
      F := 1
      R := 1

пока R <= N, повторять
НЦ
    F := F * R
    R := R + 1
кц
ВЫВОД F
кОН
  
```

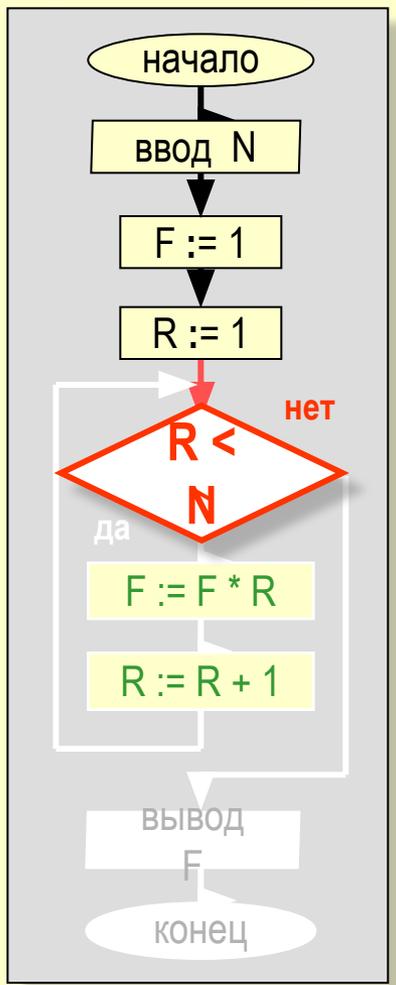


№	Операция	N	F	R	Условие
12	R := R + 1			4	

Значение переменной R увеличивается на единицу ($R := R + 1 = 3 + 1 = 4$).
 Новое значение множителя (переменной R) равно 4.

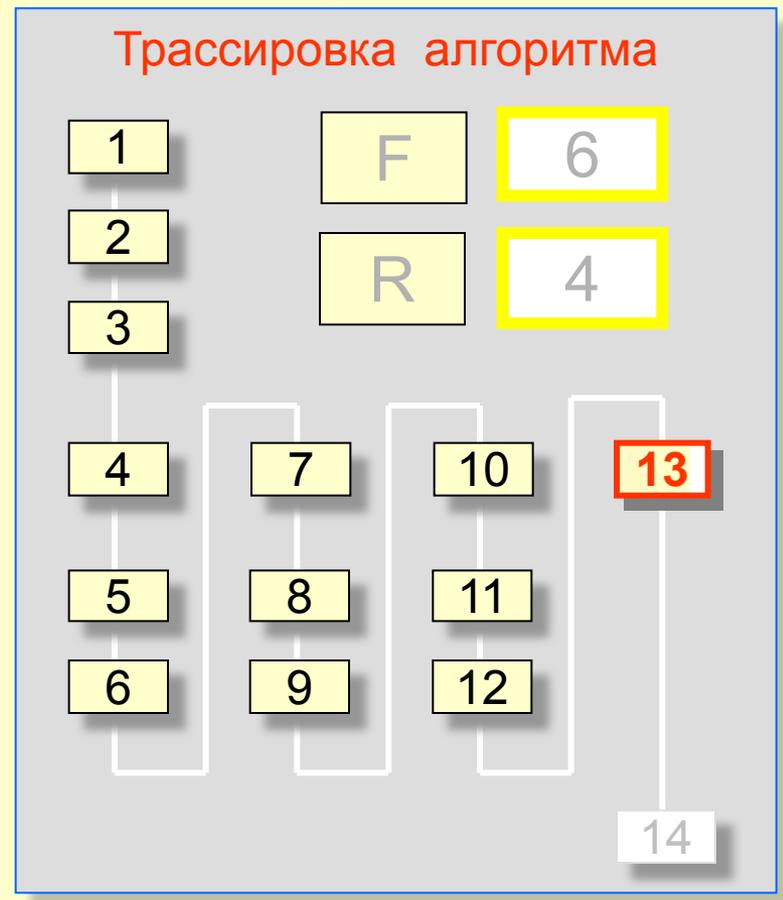


Трассировка циклического алгоритма



```

алг N!
цел F, N, R
нач  ВВОД N      N = 3
      F := 1
      R := 1
      пока R <= N, повторять
      нц
          F := F * R
          R := R + 1
      кц
      вывод F
кон
  
```

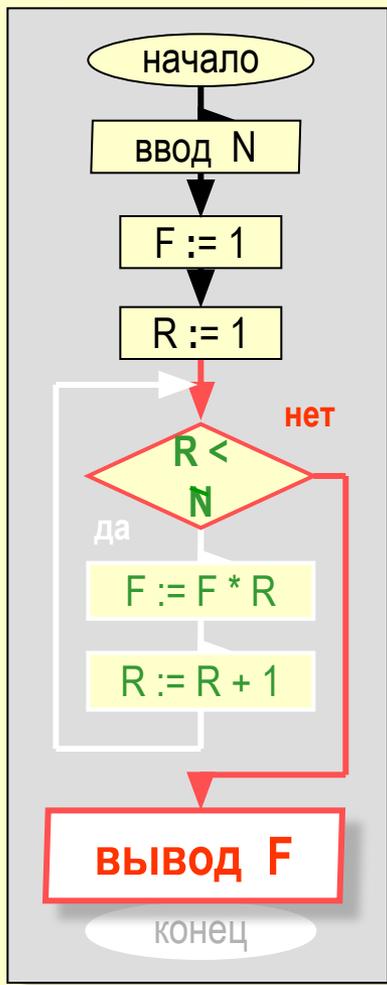


№	Операция	N	F	R	Условие
13	R <= N				4 <= 3 нет

В четвертый раз проверяется условие выполнения цикла $R \leq N$ при новом значении R . Условие на этом шаге **не выполняется**, так как $R = 4$, а $N = 3$ ($4 > 3$). После этого шага цикл заканчивается и команды тела цикла исполняться больше не могут.



Трассировка циклического алгоритма

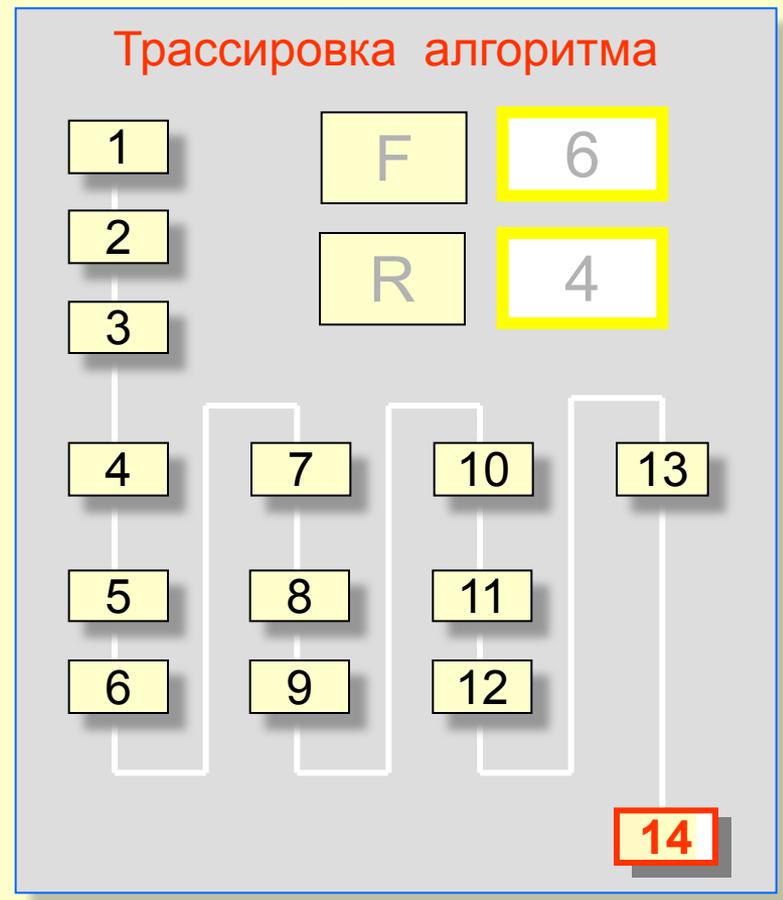


```

алг N!
цел F, N, R
нач ввод N      N = 3
                F := 1
                R := 1

пока R <= N, повторять
нц
    F := F * R
    R := R + 1
кц

вывод F
кон
  
```

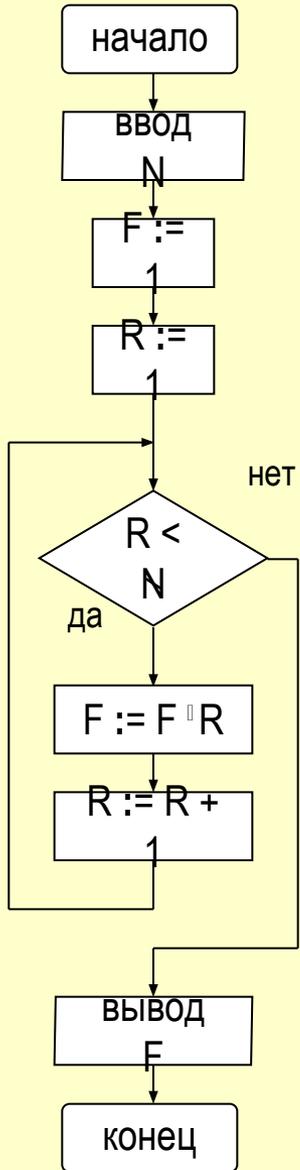


№	Операция	N	F	R	Условие
14	Вывод F		6		

Так как на шаге № 13 алгоритма вычисления факториала условие выполнения цикла не выполнено, цикл заканчивается, команды тела цикла не исполняются и управление на шаге № 14 передается команде вывода конечного значения функции F, в которой накоплено значение, равное числу 6. Это последний шаг вычислительного алгоритма.



"N - факториал" на Паскале



«N - факториал»

$$F = N! = 1 \cdot 2 \cdot \dots \cdot N$$

алг Факториал

цел F, N, R

нач

ВВОД N

F := 1

R := 1

пока R <= N, **повторять**

нц

F := F * R

R := R + 1

кц

ВЫВОД F

кон

```
PROGRAM Word ;
  M      S      ;
VAR     F, N, R : INTEGER ;
BEGIN
  N
  WRIT   ( 'Введите число букв' ) ;
  READL ( N ) ;
  F := 1 ;
  R := 1 ;
  WHILE R <= N DO
    BEGIN
      F := F * R ;
      R := R + 1 ;
    END ;
  WRITE LN ( 'Из ', N, ' букв можно
              составить ', F,
              ' слов' ) ;
END .
```

Этапы решения расчетных задач на компьютере



Постановка задачи

На этапе постановки задачи необходимо четко определить, **что дано** и **что требуется найти**

Математическая формализация

Задача переводится на язык математических формул, уравнений, отношений. Если решение задачи требует математического описания какого-то реального объекта, явления или процесса, то формализация равносильна получению соответствующей **математической модели**.

Построение алгоритма

На этапе построения алгоритма используются графические блок-схемы и Алгоритмический язык (АЯ)

Составление программы
на языке программирования

Существует множество языков и систем программирования. Чтобы составить программу, необходимо знать соответствующий формальный язык. В базовом курсе информатики для примера предлагается язык программирования ПАСКАЛЬ.

Отладка
и тестирование программы

Под отладкой программы понимается процесс испытания работы программы и исправление обнаруженных при этом ошибок. Проверка на компьютере правильности алгоритма производится с помощью тестов. Тест – это конкретный вариант значений исходных данных, для которого известен ожидаемый результат.

Проведение расчетов и анализ полученных результатов