

# Циклы

# Циклы

---

- **Цикл** – это многократное выполнение одинаковой последовательности действий.
  - цикл с известным числом шагов
  - цикл с неизвестным числом шагов (цикл с условием)
- Примеры:
  - Вывести на экран значение квадратов и кубов чисел от 1 до 10
  - Определить количество цифр в заданном числе

# Циклы

---

- Если количество повторений известно, то используются циклы с параметром:
  - **FOR ... TO ... DO**
  - **FOR ... DOWNTO ... DO**
- В противном случае:
  - цикл с предусловием – **WHILE ... DO**
  - цикл с постусловием – **REPEAT ... UNTIL.**

# Циклы

---

- При работе с операторами цикла можно использовать следующие процедуры:

- **Break**

- **Continue**

- **Break** – позволяет досрочно выйти из цикла
- **Continue** – позволяет перейти к новой итерации цикла, даже если предыдущая итерация еще не завершена

**ЦИКЛ FOR**

# Цикл с переменной

---

## Увеличение параметра на 1:

```
for <параметр цикла> := <начальное значение>  
to  
    <конечное значение> do  
begin  
    { тело цикла }  
end;
```

## Уменьшение параметра на 1:

```
for <параметр цикла> := <начальное значение>  
downto <конечное значение> do  
begin  
    { тело цикла }  
end;
```

# ЦИКЛЫ

---

**Пример 1:** Вывести на экран квадраты и кубы чисел от 1 до 10

```
var i:byte;  
begin  
  for i:=1 to 10 do  
    begin  
      WriteLn(i:3, sqr(i):4, i*i*i:5);  
    end;  
  readln;  
end.
```

# Циклы с параметром

---

## Особенности:

- переменная цикла может быть только целой (`integer`, `byte`, `word` и т.п.)
- шаг изменения переменной цикла всегда равен 1 (`to`) или -1 (`downto`)
- если в теле цикла только один оператор, слова `begin` и `end` можно не писать:

```
for i:=1 to 10 do  
    WriteLn(i:3, sqr(i):4, i*i*i:5);
```

- если конечное значение меньше начального, цикл (`to`) не выполняется ни разу (проверка условия в начале цикла, цикл с предусловием)



# Циклы с параметром

---

## Особенности:

- в теле цикла не разрешается изменять переменную цикла

```
n := 8;  
for i:=1 to n do begin  
    writeln('Привет');  
    i := i + 5;  
end;
```

**Запрещено**, т.к. параметр *i* может не достигнуть значения *n*.

*Зацикливание*

# Циклы с параметром

---

## Особенности:

- Начальное и конечное значения цикла определяются заранее, поэтому при изменении начального и конечного значения внутри цикла количество шагов не изменится:

```
n := 8;  
for i:=1 to n do begin  
    writeln('Привет');  
    n := n + 1;  
end;
```

нет  
зацикливани  
я

# Циклы с параметром

## Особенности:

- после выполнения цикла во многих системах устанавливается первое значение переменной цикла, при котором нарушено условие

```
for i:=1 to 8 do  
    writeln('Привет');  
writeln('i=', i);
```

i=9

```
for i:=8 downto 1 do  
    writeln('Привет');  
writeln('i=', i);
```

i=0

# Определите значение

## переменной $a$

```
a := 1;  
for i := 1 to 3 do a := a + 1;
```

$a = 4$

```
a := 1;  
for i := 3 to 1 do a := a + 1;
```

$a = 1$

```
a := 1;  
for i := 1 downto 3 do a := a + 1;
```

$a = 1$

```
a := 1;  
for i := 3 downto 1 do a := a + 1;
```

$a = 4$

# Как изменить шаг?

**Задача.** Вывести на экран квадраты и кубы нечётных целых чисел от 1 до 9.

**Особенность:** переменная цикла должна увеличиваться на 2.

**Решение:**

...

```
for i:=1 to 9 do begin
```

```
  if i mod 2 = 1 then begin
```

```
    i2 := i*i;
```

```
    i3 := i2*i;
```

```
    writeln(i:4, i2:4, i3:4);
```

```
  end;
```

```
end;
```

...

выполняется  
только  
для  
нечётных  $i$



Что плохо?

# Как изменить шаг? – II

---

**Идея:** Надо вывести всего 5 чисел, переменная  $k$  изменяется от 1 до 5. Начальное значение  $i$  равно 1, с каждым шагом цикла  $i$  увеличивается на 2.

**Решение:**

```
i := 1;
```

???

```
for k:=1 to 5 do begin
```

```
  i2 := i*i;
```

```
  i3 := i2*i;
```

```
  i := i + 2; i2:4, i3:4);
```

???

```
end;
```

...

# Как изменить шаг? – III

**Идея:** Надо вывести всего 5 чисел, переменная  $k$  изменяется от 1 до 5. **Зная**  $k$ , **надо рассчитать**

$k$	1	2	3	4	5
$i$	1	3	5	7	9

$i = 2k - 1$

**Решение:**

...

```
for  $i := 2 * k - 1;$  begin
```

???

```
  i2 := i * i;
```

```
  i3 := i2 * i;
```

```
  writeln(i:4, i2:4, i3:4);
```

```
end;
```

...

# Цикл с параметром

---

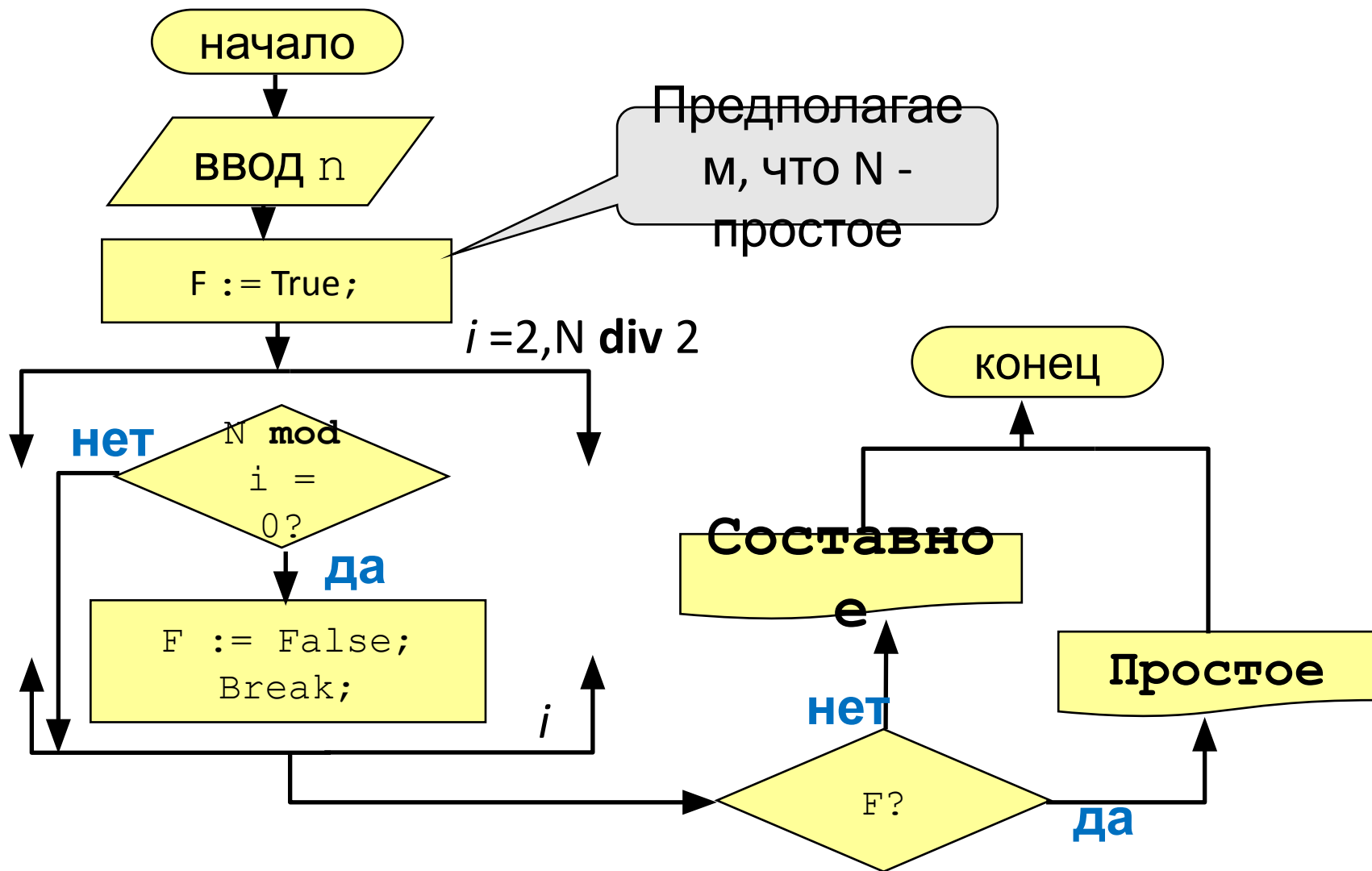
**Задача:** Определить, является ли число  $N$  простым.

**Идея:** Если у числа есть делители, отличные от 1 и  $N$ , то число  $N$  – составное, иначе - простое

Для обозначения наличия делителей у числа, зададим переменную-флаг, которая будет принимать значение **true** (простое) или **false** (составное)



# Алгоритм



# Цикл с параметром

---

## Программа:

```
var i, N:byte;
      F : Boolean;
begin
  Write( 'Введите N:' ); ReadLn(N);
  F := True;
  for i := 2 to N div 2 do
    if N mod i = 0 then begin
      F := false;
      break;
    end;
  if F then write( 'Простое' )
    else write( 'Составное' );
  readln;
end.
```

# Цикл с параметром

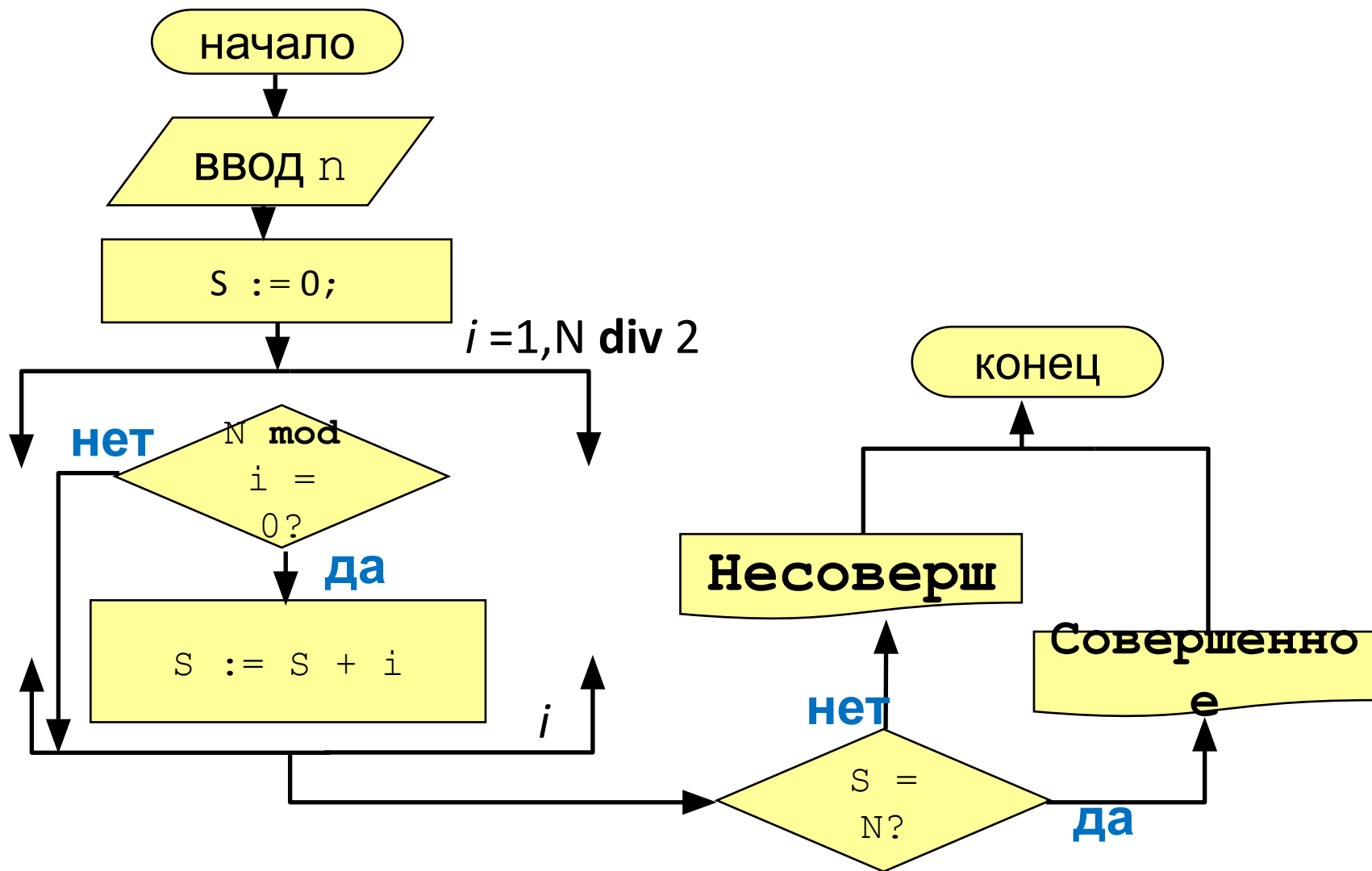
---

**Задача:** Определить, является ли число  $N$  совершенным. Совершенное число равно сумме своих делителей (кроме самого числа  $N$ )

Пример:  $6 = 1 + 2 + 3$

**Идея:** 1. Определить сумму делителей числа  $N$   
2. Сравнить сумму с  $N$ , если равны – совершенное число, иначе – нет

# Алгоритм



# Цикл с параметром

---

## Программа:

```
var i, N:byte;  
      S : word;  
begin  
  Write( 'Введите N:' ); ReadLn(N);  
  S := 0;  
  for i := 1 to N div 2 do  
    if N mod i = 0 then S := S + i;  
  if S = N then write( 'Совершенное' )  
    else write( 'Несовершенное' );  
  readln;  
end.
```

# Цикл с параметром

---

**Задача:** Определить, количество совершенных чисел в диапазоне от 1 до 1000

**Идея:** 1. Запустить цикл от 1 до 1000

2. Для каждого из чисел диапазона, выполнить шаги предыдущего алгоритма

3. Если число – совершенное, увеличить счетчик на 1

# Цикл с параметром

---

## Программа:

```
var i, N:1..1000;  
      S , k: word;  
Begin  
  k := 0;  
  for N := 1 to 1000 do  
    begin  
      S := 0;  
      for i := 1 to N div 2 do  
        if N mod i = 0 then S := S + i;  
        if S = N then inc(k);  
      end;  
  WriteLn( 'Количество совершенных чисел =' , k );  
  readln;  
end.
```

# **ЦИКЛЫ С УСЛОВИЕМ**



# Циклы с неизвестным числом шагов

**Задача:** Ввести целое число ( $< 2000000$ ) и определить количество цифр в нем.

**Идея решения:** Отсекаем последовательно последнюю цифру, увеличиваем счетчик.

n	count
123	0
12	1
1	2
0	3

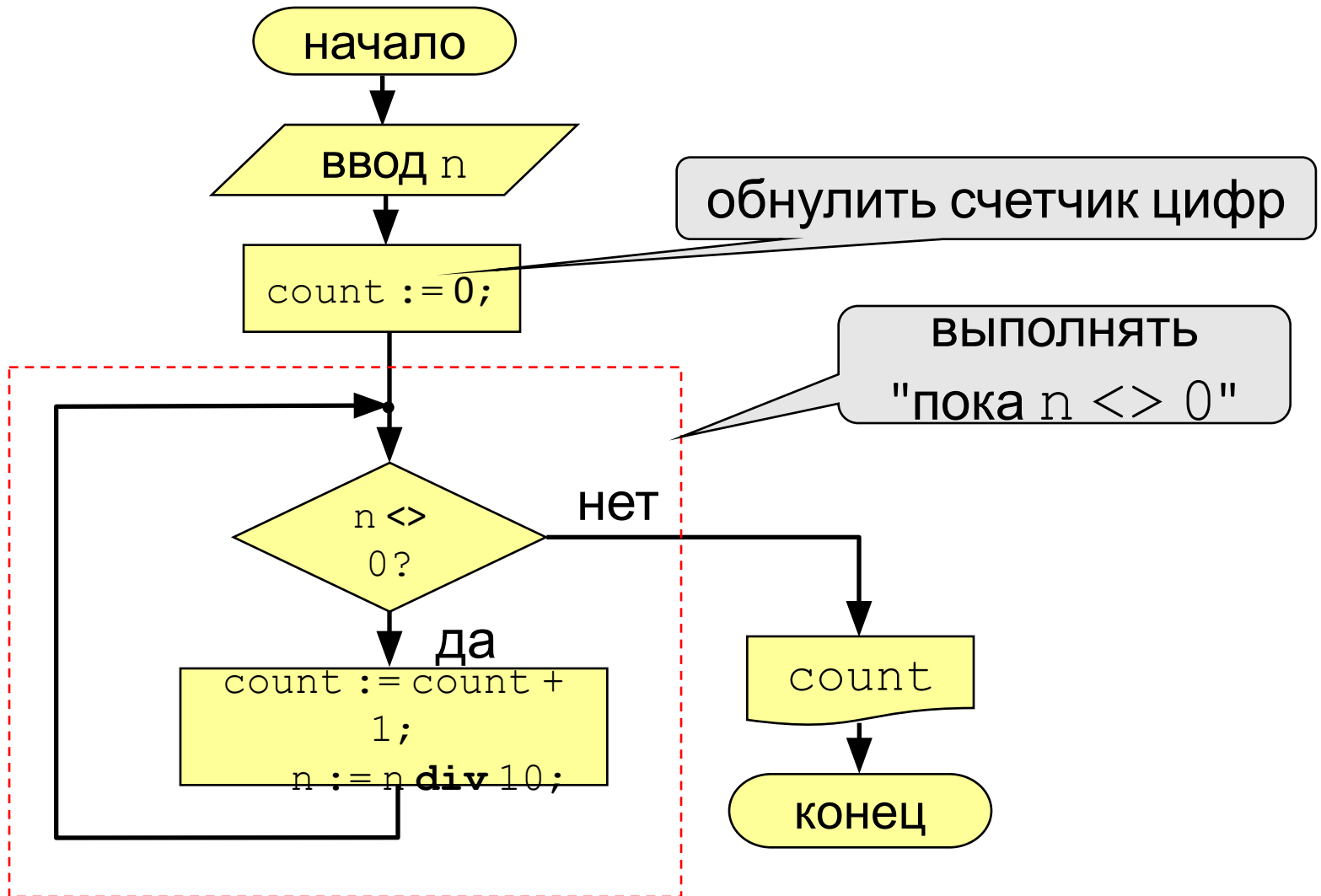
n	count
3567	0
356	1
35	2
3	3
0	4

**Проблема:** Неизвестно, сколько шагов надо сделать.

**Решение:** Надо остановиться, когда  $n = 0$ , т.е.

надо писать `while n > 0`

# Алгоритм



# Программа

```
program qq;  
var n, count, n1: integer;  
begin  
  writeln('Введите целое число');  
  read(n); n1 := n;  
  count := 0;  
  while n <> 0 do begin  
    count := count + 1;  
    n := n div 10;  
  end;  
  writeln('В числе ', n1, ' нашли ',  
    count, ' цифр');  
end.
```

ВЫПОЛНЯТЬ "пока  
n <> 0"



# Цикл с предусловием

---

```
while <условие> do begin
    {тело цикла}
end;
```

## Особенности:

- МОЖНО ИСПОЛЬЗОВАТЬ СЛОЖНЫЕ УСЛОВИЯ:

```
while (a<b) and (b<c) do begin
    {тело цикла}
end;
```

- если в теле цикла только один оператор, слова `begin` и `end` можно не писать:

```
while a < b do
    a := a + 1;
```

# Цикл с предусловием

---

## Особенности:

- условие пересчитывается каждый раз при входе в цикл
- если условие на входе в цикл ложно, цикл не выполняется ни разу

```
a := 4; b := 6;  
while a > b do  
    a := a - b;
```

- если условие никогда не станет ложным, программа зацикливается

```
a := 4; b := 6;  
while a < b do  
    d := a + b;
```

# Сколько раз выполняется цикл?

```
a := 4; b := 6;  
while a < b do a := a + 1;
```

2 раза

a = 6

```
a := 4; b := 6;  
while a < b do a := a + b;
```

1 раз

a = 10

```
a := 4; b := 6;  
while a > b do a := a + 1;
```

0 раз

a = 4

```
a := 4; b := 6;  
while a < b do b := a - b;
```

1 раз

b = -2

```
a := 4; b := 6;  
while a < b do a := a - 1;
```

зацикливани

e

# Замена `for` на `while` и наоборот

```
for i:=1 to 10 do
begin
  {тело цикла}
end;
```

```
i := 1;
while i <= 10 do
begin
  {тело цикла}
  i := i + 1;
end;
```

```
for i:=a downto b
do
begin
  {тело цикла}
end;
```

```
i := a;
while i >= b do
begin
  {тело цикла}
  i := i - 1;
end;
```

Замена цикла `for` на `while` возможна **всегда**.

Замена `while` на `for` возможна только тогда, когда можно заранее **рассчитать число шагов**

# Цикл с постусловием

---

**Задача:** Ввести **целое положительное** число (<2000000) и определить число цифр в нем.

**Проблема:** Как не дать ввести отрицательное число или ноль?

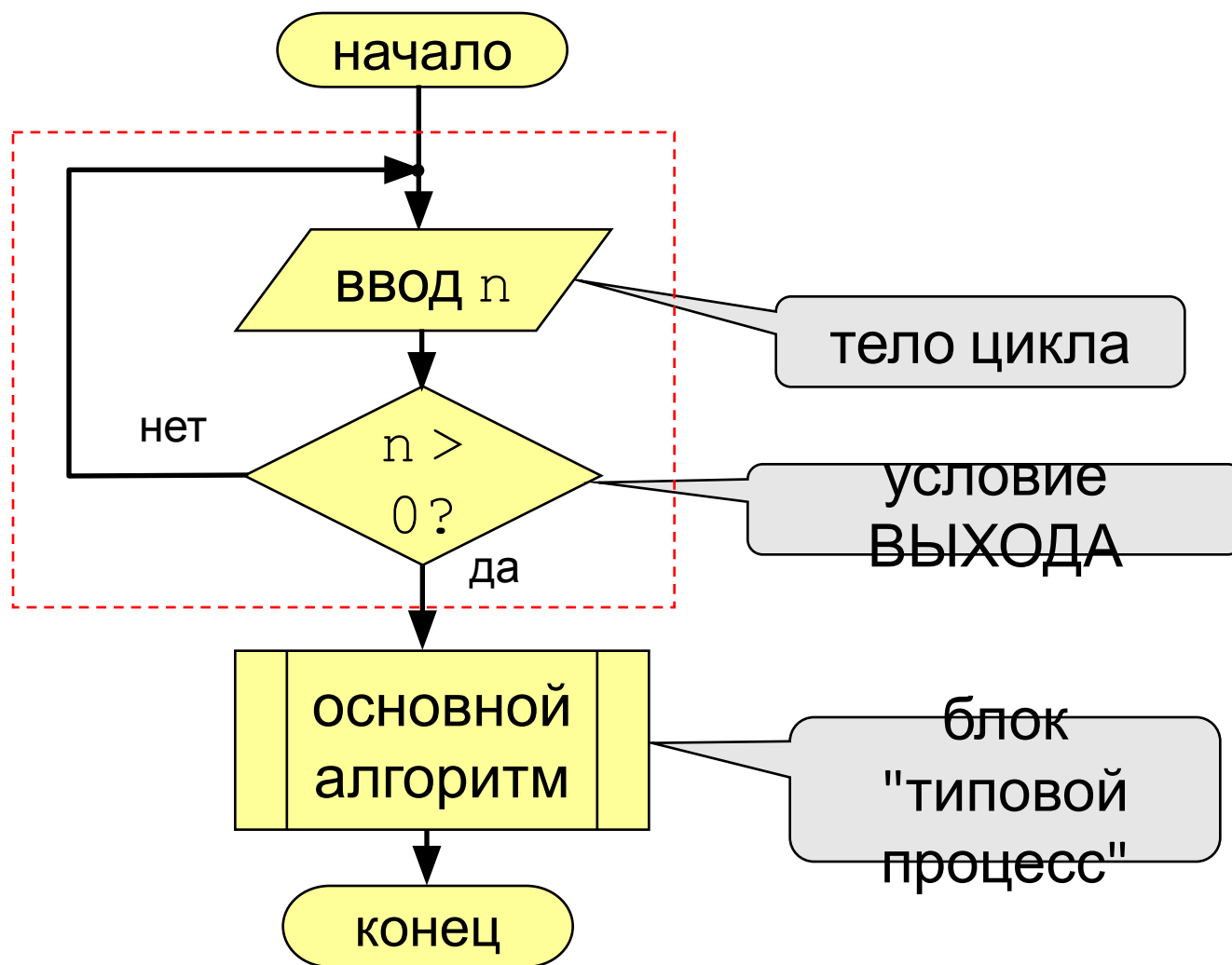
**Решение:** Если вводится неверное число, вернуться назад к вводу данных (цикл!).

**Особенность:** Один раз тело цикла нужно выполнить в любом случае => проверку условия цикла надо делать в конце цикла (цикл с постусловием).

**Цикл с постусловием** – это цикл, в котором проверка условия выполняется в конце цикла.



# Цикл с постусловием: алгоритм



# Программа

---

```
program qq;
```

```
var n: integer;
```

```
begin
```

```
repeat
```

```
  writeln('Введите положительное число');
```

```
  read(n);
```

```
until n > 0;
```

условие ВЫХОДА

```
  ... { основной алгоритм }
```

```
end.
```

# Программа

## Особенности:

- тело цикла всегда выполняется хотя бы один раз
- после слова **until** ("до тех пор, пока не...") ставится условие ВЫХОДА из цикла
- В отличие от цикла **WHILE** цикл **REPEAT** повторяется пока условие ЛОЖНО!
- Независимо от количества операторов в теле цикла, слова `begin` и `end` можно не писать

```
a := 4; b := 6;
```

```
repeat
```

```
    d := a + b;
```

```
    a := a + 1;
```

```
until a > b;
```

2 оператора в теле  
цикла.  
`begin` и `end` не ставятся

# Сколько раз выполняется цикл?

```
a := 4; b := 6;  
repeat a := a + 1; until a > b;
```

3 раза  
a = 7

```
a := 4; b := 6;  
repeat a := a + b; until a > b;
```

1 раз  
a = 10

```
a := 4; b := 6;  
repeat a := a + b; until a < b;
```

зацикливание

```
a := 4; b := 6;  
repeat b := a - b; until a < b;
```

2 раза  
b = 6

```
a := 4; b := 6;  
repeat a := a + 2; until a < b;
```

зацикливание

# Последовательности

## Примеры:

- 1, 2, 3, 4, 5, ...

$$a_n = n$$

$$a_1 = 1, \quad a_{n+1} = a_n + 1$$

- 1, 2, 4, 7, 11, 16, ...

$$a_1 = 1, \quad a_{n+1} = a_n + n$$

- 1, 2, 4, 8, 16, 32, ...

$$a_n = 2^{n-1}$$

$$a_1 = 1, \quad a_{n+1} = 2a_n$$

- $\frac{1}{2}, \frac{1}{2}, \frac{3}{8}, \frac{1}{4}, \frac{5}{32}, \dots$

- $\frac{1}{2}, \frac{2}{4}, \frac{3}{8}, \frac{4}{16}, \frac{5}{32}, \dots$

$$a_n = \frac{b_n}{c_n}$$

$$b_1 = 1, \quad b_{n+1} = b_n + 1$$

$$c_1 = 2, \quad c_{n+1} = 2c_n$$

# Последовательности

**Задача 1:** найти сумму всех элементов последовательности,

$$1, -\frac{1}{2}, \frac{2}{4}, -\frac{3}{8}, \frac{4}{16}, -\frac{5}{32}, \dots$$

которые по модулю больше 0,001:

$$S = 1 - \frac{1}{2} + \frac{2}{4} - \frac{3}{8} + \frac{4}{16} - \frac{5}{32} + \dots$$

Элемент последовательности (начиная с №2):

$$a = z \frac{b}{c}$$

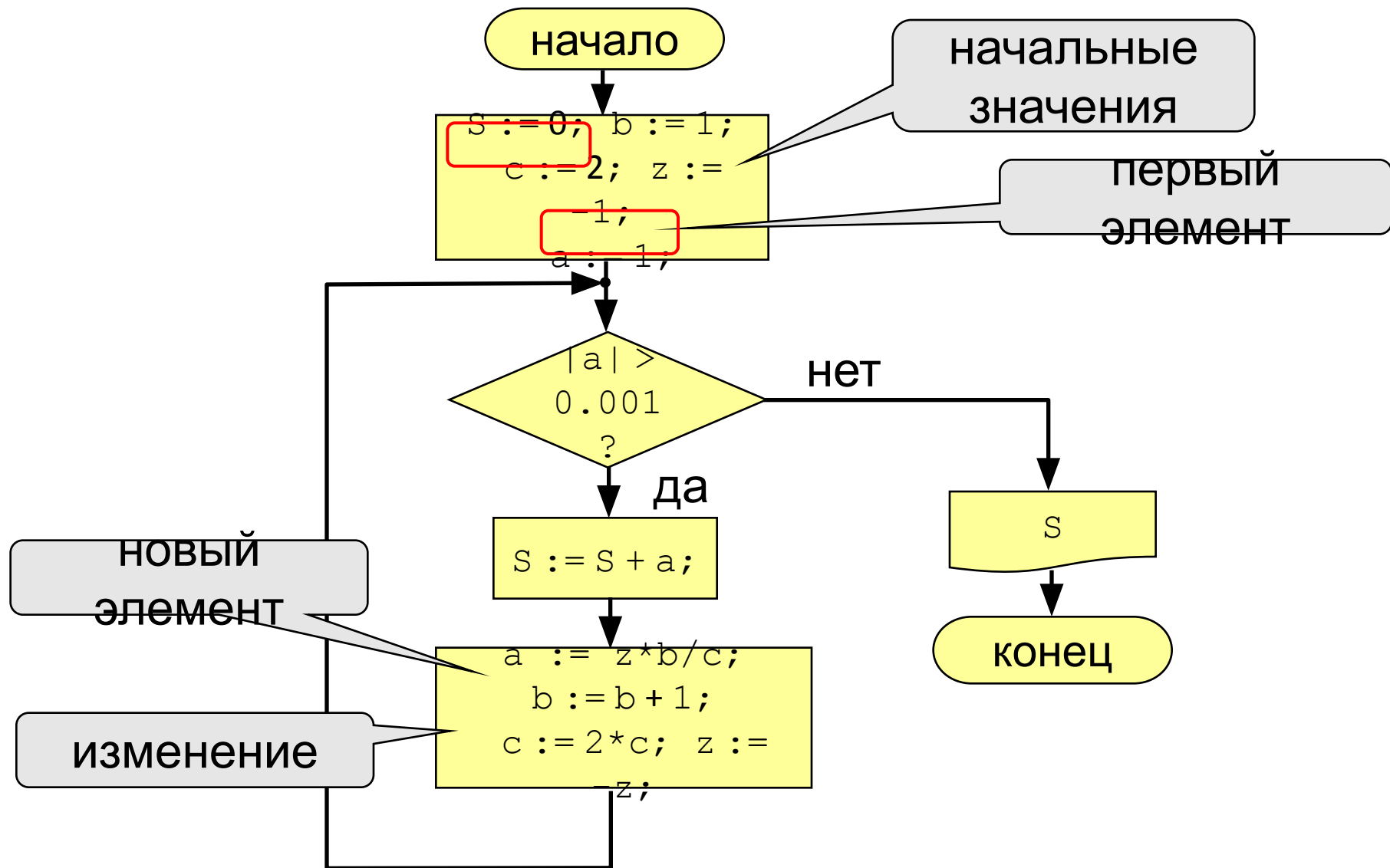
<b>n</b>	1	2	3	4	5	...
<b>b</b>	1	2	3	4	5	...
<b>c</b>	2	4	8	16	32	...
<b>z</b>	-1	1	-1	1	-1	...

`b := b+1;`

`c := 2*c;`

`z := -z;`

# Алгоритм



# Программа

---

```
program qq;  
var b, c, z: integer;  
    S, a: real;  
begin  
    S := 0; z := -1;  
    b := 1; c := 2; a := 1;  
    while abs(a) > 0.001 do begin  
        S := S + a;  
        a := z * b / c;  
        z := - z;  
        b := b + 1;  
        c := c * 2;  
    end;  
    writeln('S =', S:10:3);  
end.
```

начальные  
значения

увеличение  
суммы

расчет элемента  
последовательности

переход к  
следующему  
слагаемому



# Последовательности

---

## Задача 2:

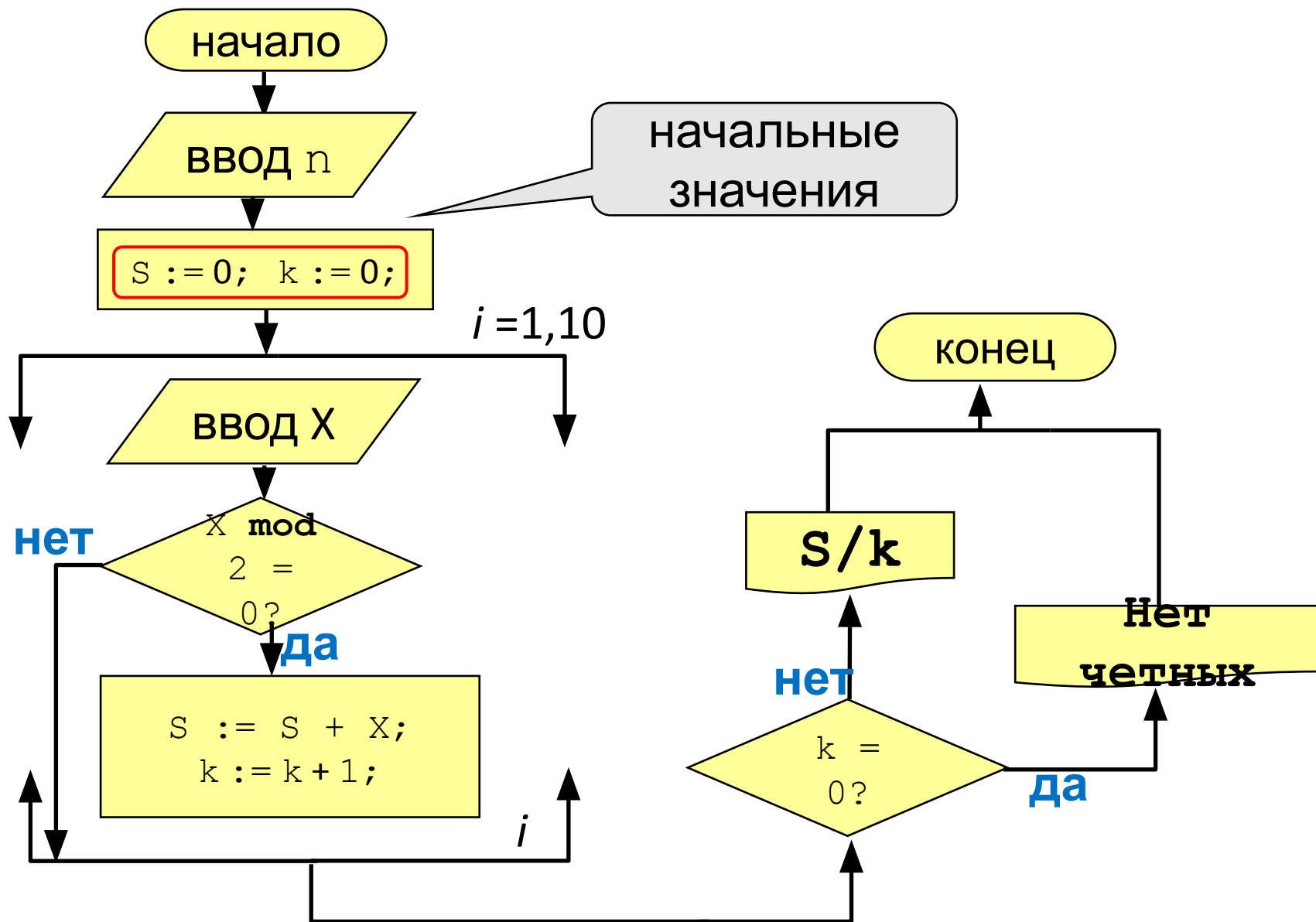
Дана последовательность из  $N$  целых чисел.

Требуется вычислить среднее арифметическое четных чисел последовательности.

**Особенность:** Сначала определяем  $N$ . Затем вводим с клавиатуры заданное количество чисел.

Для подсчета среднего арифметического потребуется вычисление суммы и количества четных элементов

# Алгоритм



# Последовательности

## Программа:

```
var i, N, k, X:byte;
    s: integer;
begin
  Write('Введите N:');ReadLn(N);
  s := 0; k := 0;
  for i:=1 to N do begin
    Read(X);
    if x mod 2 = 0 then
      begin inc(s,X); inc(k); end;
    end;
  if k=0 then write('Нет четных чисел') else
    write('Среднее арифм.=' , s/k);
    readln;
end.
```

# Последовательности

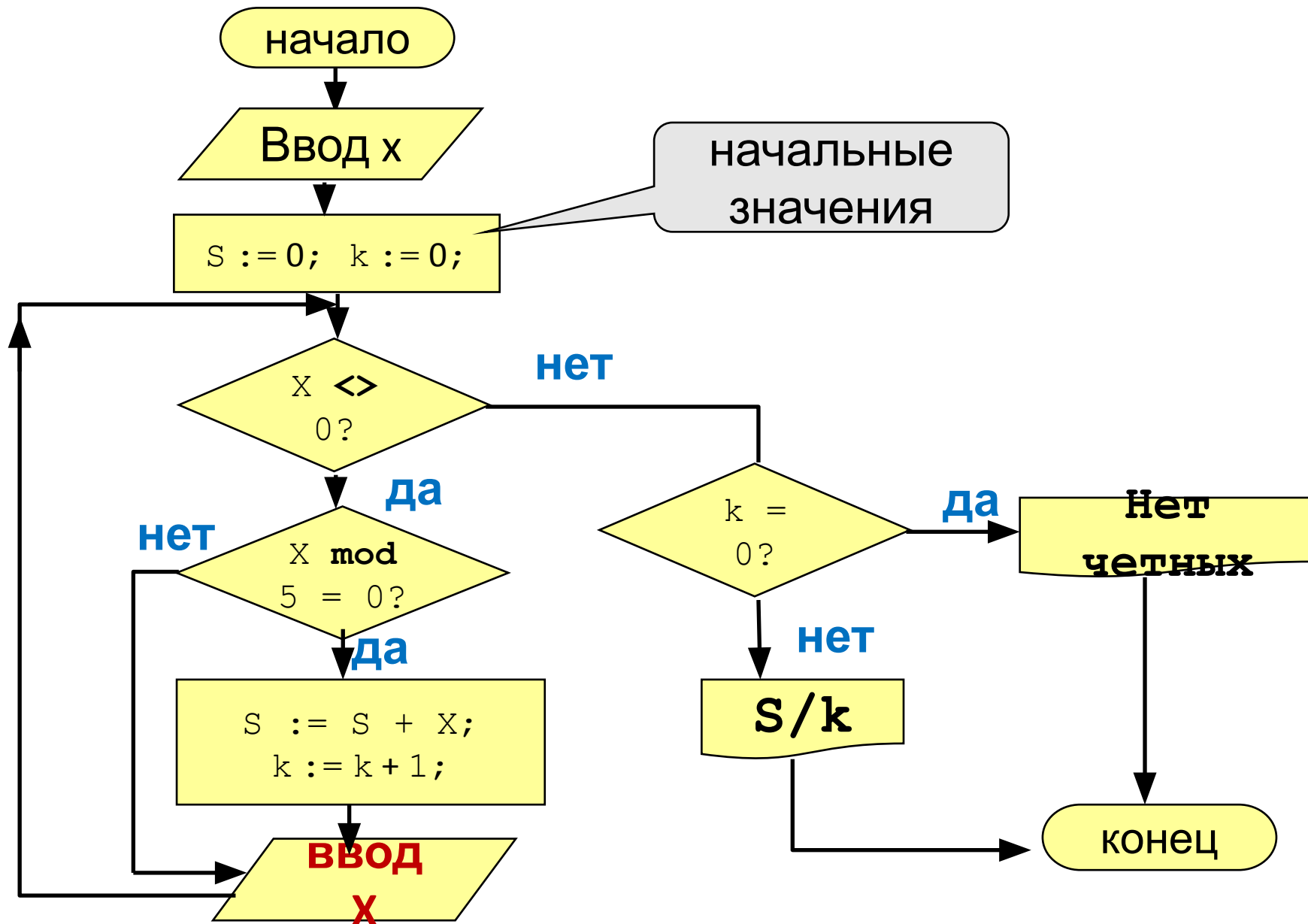
---

## Задача 3:

Дана последовательность целых чисел, за которой следует ноль. Требуется вычислить среднее арифметическое кратных 5 элементов последовательности.

**Особенность:** Заранее неизвестно количество чисел последовательности.

# Алгоритм



# Последовательности

---

## Программа:

```
var i, N, k, X:byte;
      s: integer;
begin
  Write('Введите X:');ReadLn(X);
  s := 0; k := 0;
  while x<>0 do begin
    if x mod 5 = 0 then begin
      inc(s,X); inc(k); end;
    Read(X);
  end;
  if k=0 then write('Нет четных чисел') else
    write('Среднее арифм.=' , s/k);
  readln;
end.
```