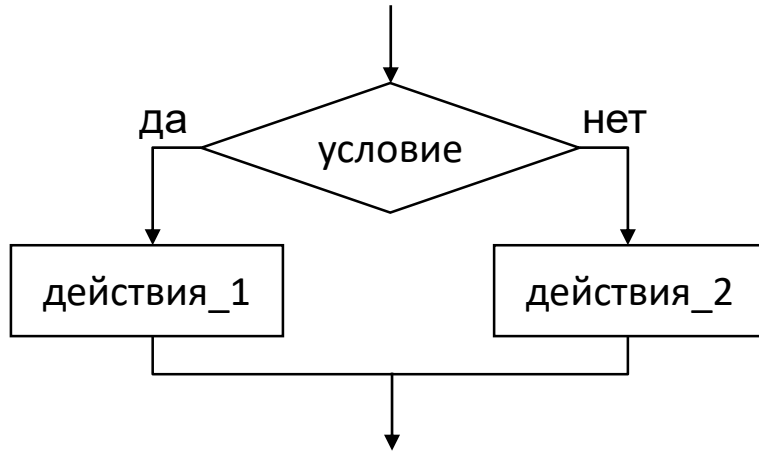


Основные управляющие конструкции языка Python

Операторы ветвления

Полная форма ветвления (условного оператора):



```
if <условие>:  
    <блок_операторов_1>  
else:  
    <блок_операторов_2>
```

Если условие истинно, то выполняется блок_1, **иначе** – блок_2.

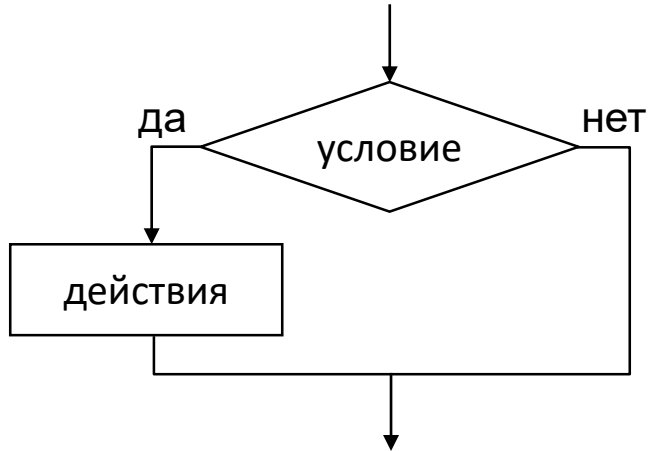
*В языке Python важную роль играют отступы операторов от левой границы текста программы. Начало и конец блоков операторов определяется этими сдвигами. Задаются отступы пробелами или клавишей **Tab**.*

Если в блоке всего один оператор, то допустима такая форма записи:

```
if <условие>: <оператор_1>  
else: <оператор_2>
```

Операторы ветвления

Неполная форма ветвления (условного оператора):



```
if <условие>:  
    <блок_операторов>
```

Если условие истинно, то выполняется блок операторов.
В противном случае – переход к следующему оператору программы.

Если в блоке всего один оператор, то допустима такая форма записи:

```
if <условие>: <оператор>
```

Составные условия

В качестве условий можно использовать как простые логические выражения, так и составные, записанные с помощью логических операций.

- **Логическое умножение `and` («И»)**. Составное условие истинно, если истинны все простые условия, входящие в него.

`5>3 and 2*2=4` **#истинно**

`5>3 and 2*2=5` **#ложно**

- **Логическое сложение `or` («ИЛИ»)**. Составное условие истинно, если истинно хотя бы одно простое условие, входящее в него.

`5<3 or 2*2=4` **#истинно**

`5<3 or 2*2=5` **#ложно**

- **Логическое отрицание `not` («НЕ»)**. Если условие истинно, то его отрицание ложно, и наоборот.

`not 2*2=4` **#ложно**

Приоритет операций:

1) `<`, `>`, `<=`, `>=`, `==`, `!=`

2) `not`

3) `and`

4) `or`

Вложенные ветвления

Внутри условного оператора могут находиться любые операторы, в том числе и другие условные операторы (**вложенные условные операторы**).

Это позволяет сделать выбор не из двух, а из нескольких вариантов.

Если после **else** следует еще один оператор **if**, можно использовать «каскадное» ветвление с ключевыми словами **elif** («else-if»).

Если очередное условие ложно, то выполняется проверка следующего условия и т. д.

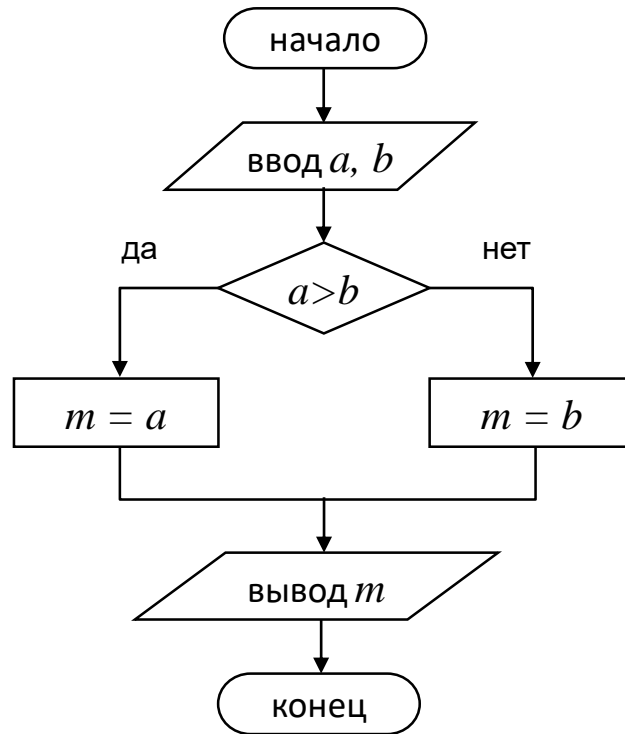
```
if <условие_1>:  
    <блок_1>  
else:  
    if <условие_2>:  
        <блок_2>  
    else:  
        <блок_3>
```

```
if <условие_1>:  
    <блок_1>  
elif <условие_2>:  
    <блок_2>  
    . . .  
else:  
    <блок_N>
```

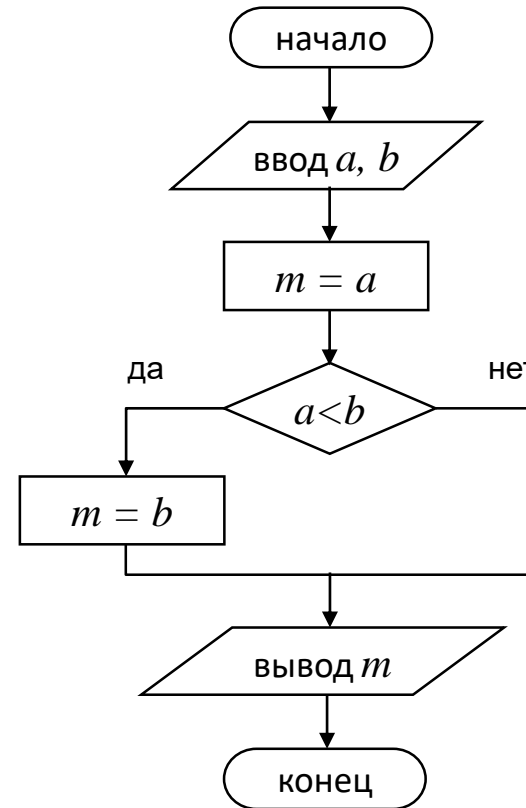
Задача 1

Определить большее из двух значений переменных.

1 способ



2 способ



Задача 1

Определить большее из двух значений переменных.

1 способ

```
# Больше из двух
print ("Введите два числа:")
a = int(input("a="))
b = int(input("b="))
if a>b: m = a
else: m = b
print ("Большее число", m)
```

```
Введите два числа:
a=5
b=6
Большее число 6
```

2 способ

```
# Больше из двух
print ("Введите два числа:")
a = int(input("a="))
b = int(input("b="))
m = a
if a<b: m = b
print ("Большее число", m)
```

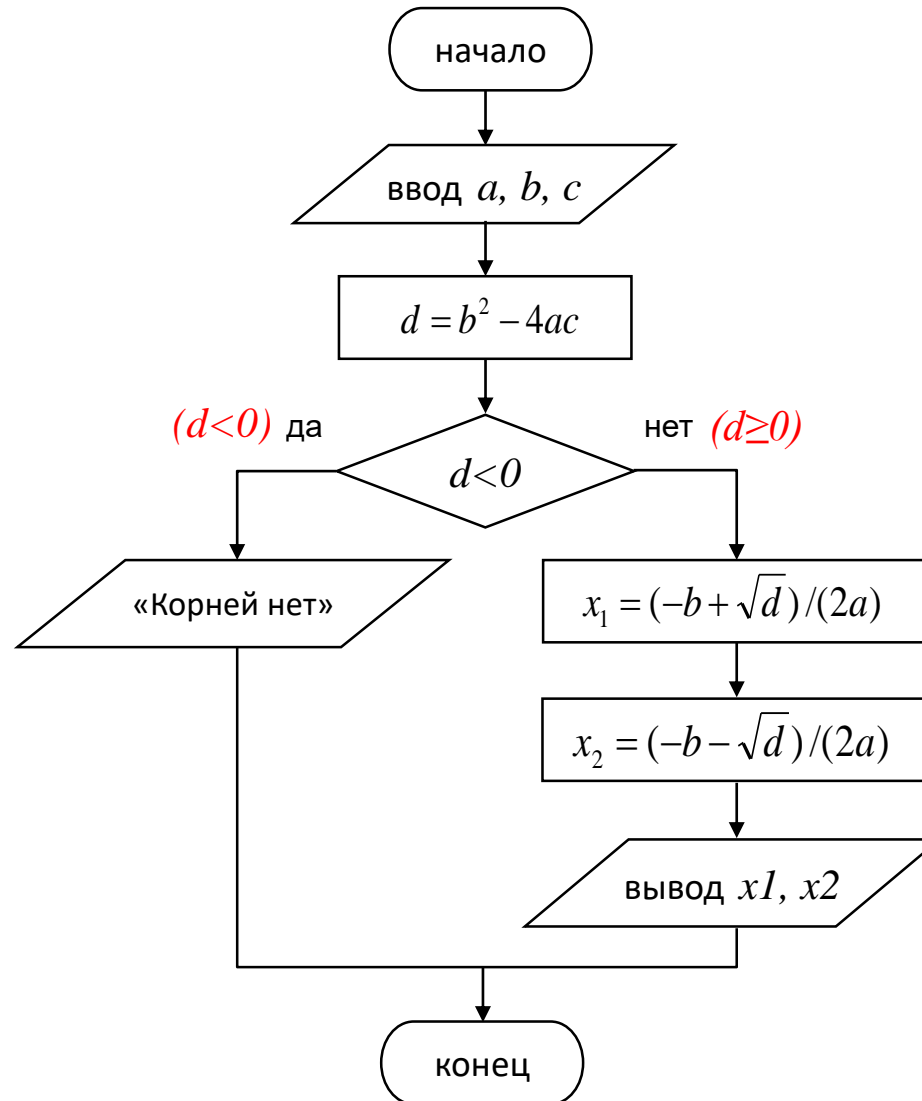
```
Введите два числа:
a=6
b=5
Большее число 6
```

Примечание: в языке Python есть встроенная функция max.

```
m = max(a, b)
```

Задача 2

Найти корни данного квадратного уравнения $ax^2+bx+c=0$.



Задача 2

Найти корни данного квадратного уравнения $ax^2+bx+c=0$.

```
# Решение квадратного уравнения
from math import *
print ("Введите коэффициенты a, b, c:")
a = float(input("a= "))
b = float(input("b= "))
c = float(input("c= "))
d = b**2 - 4*a*c          # дискриминант
if d<0: print ("Корней нет!")
else:
    x1 = (-b + sqrt(d)) / (2 * a)
    x2 = (-b - sqrt(d)) / (2 * a)
    print ("x1=", "{:6.2f}".format(x1))
    print ("x2=", "{:6.2f}".format(x2))
```

Введите коэффициенты a, b, c:

a= 1

b= 3

c= 2

x1= -1.00

x2= -2.00

Введите коэффициенты a, b, c:

a= 4

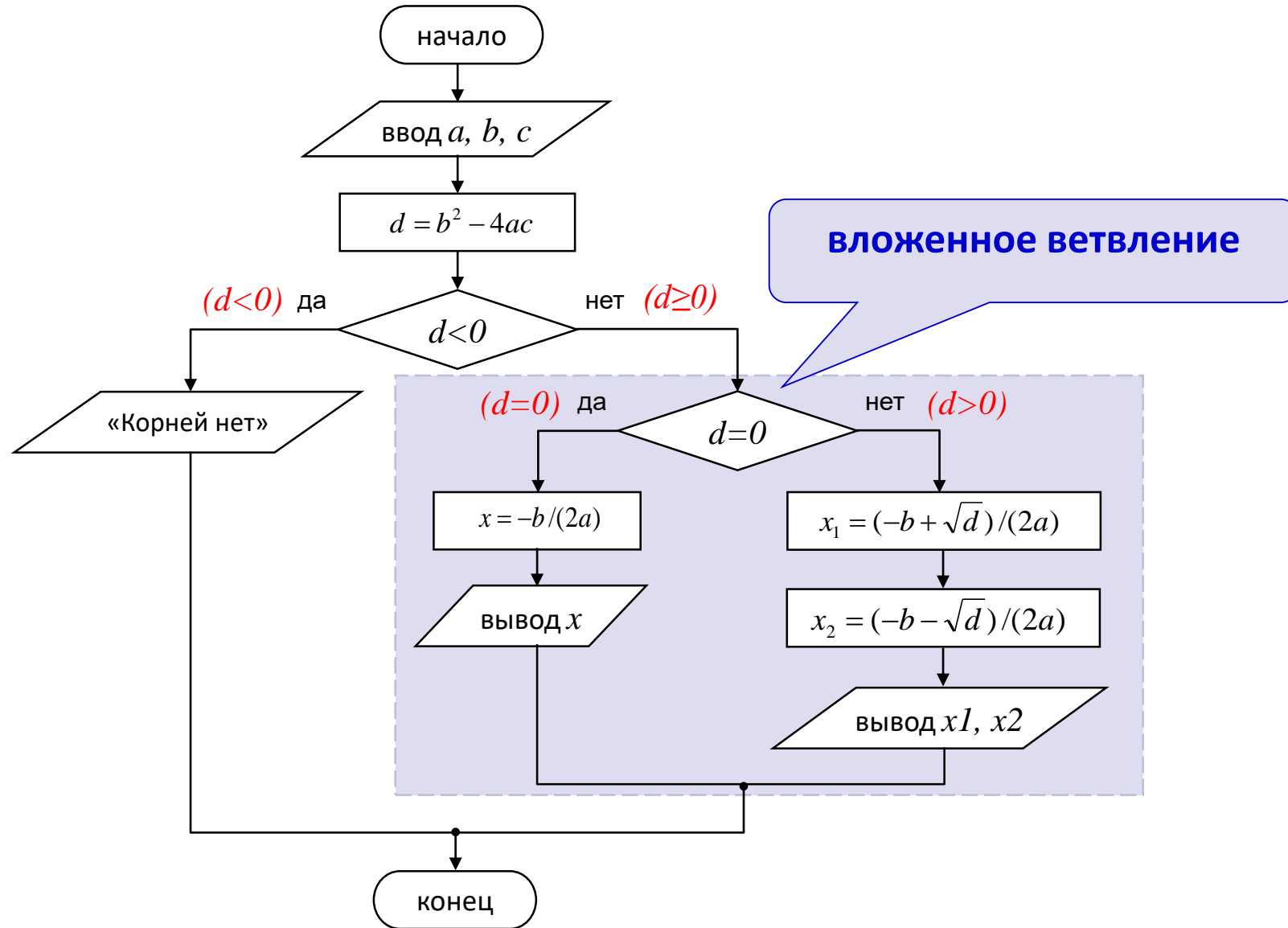
b= 1

c= 4

Корней нет!

Задача 3

Найти корни данного квадратного уравнения $ax^2+bx+c=0$, учитывая случай, когда уравнение имеет один корень.



Задача 3

Найти корни данного квадратного уравнения $ax^2+bx+c=0$, учитывая случай, когда уравнение имеет один корень.

```
# Решение квадратного уравнения (ввод)
from math import *
print ("Введите коэффициенты a, b, c:")
a = float(input("a= "))
b = float(input("b= "))
c = float(input("c= "))
d = b**2 - 4*a*c

if d<0:
    print ("Корней нет!")
else:
    if d==0:
        x = -b/(2*a)
        print ("Корень один x=", "{:6.2f}".format(x))
    else:
        x1 = (-b + sqrt(d)) / (2 * a)
        x2 = (-b - sqrt(d)) / (2 * a)
        print ("Два корня:")
        print ("x1=", "{:6.2f}".format(x1))
        print ("x2=", "{:6.2f}".format(x2))
```

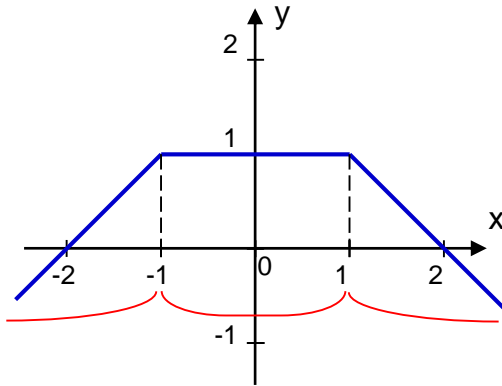
Введите коэффициенты a, b, c:
a= 4
b= 1
c= 4
Корней нет!

Введите коэффициенты a, b, c:
a= 4
b= 4
c= 1
Корень один x= -0.50

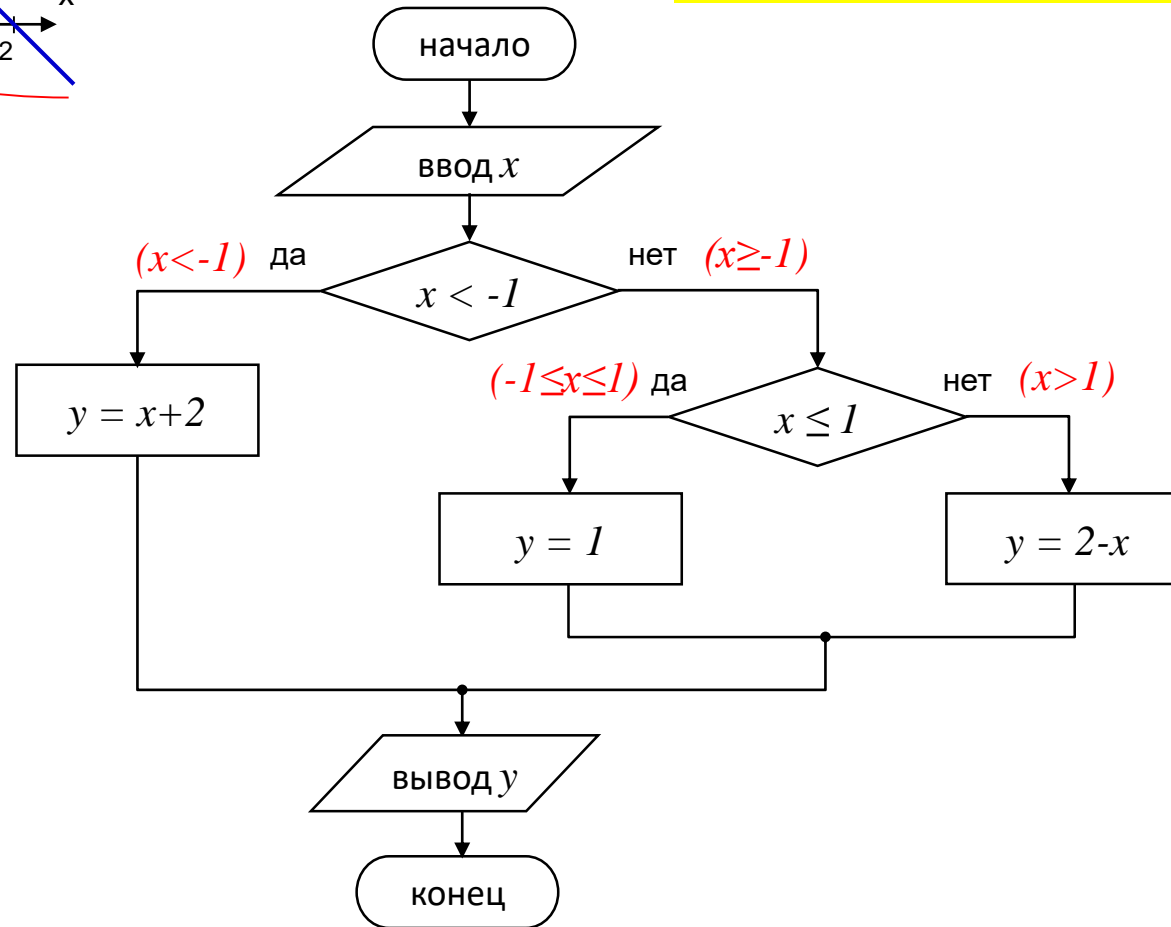
Введите коэффициенты a, b, c:
a= 1
b= 3
c= 2
x1= -1.00
x2= -2.00

Задача 4 (1 способ)

Найти значение функции для любого значения аргумента.



$$y = \begin{cases} x + 2, & \text{при } x < -1 \\ 1, & \text{при } -1 \leq x \leq 1 \\ 2 - x, & \text{при } x > 1 \end{cases}$$



Задача 4 (1 способ)

Найти значение функции для любого значения аргумента.

$$y = \begin{cases} x+2, & \text{при } x < -1 \\ 1, & \text{при } -1 \leq x \leq 1 \\ 2-x, & \text{при } x > 1 \end{cases}$$

```
# Значение функции
x = float(input("Введите x: "))
if x < -1:
    y = x+2
else:
    if x <= 1:
        y = 1
    else:
        y = 2-x
print ("y=", y)
```

Введите x: -2
y= 0.0

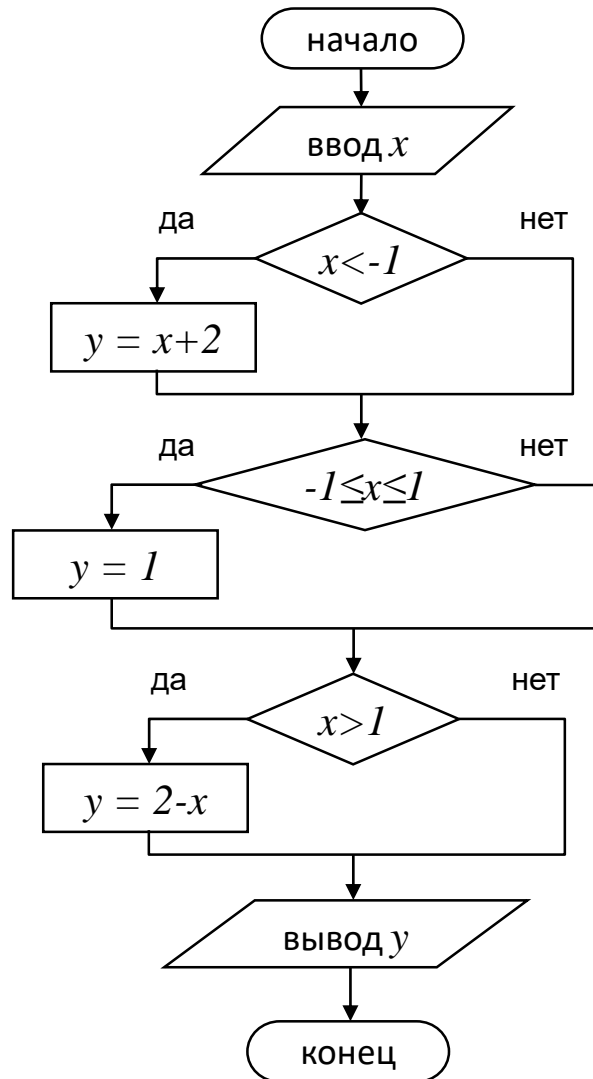
Введите x: 0.5
y= 1

Введите x: 2
y= 0.0

Задача 4 (2 способ)

Найти значение функции для любого значения аргумента.

$$y = \begin{cases} x+2, & \text{при } x < -1 \\ 1, & \text{при } -1 \leq x \leq 1 \\ 2-x, & \text{при } x > 1 \end{cases}$$



```
# Значение функции
x = float(input("Введите x: "))
if x < -1: y = x+2
if x >= -1 and x <= 1: y = 1
if x > 1: y = 2-x
print("y=", y)
```

Примечание: в языке Python разрешены двойные неравенства, например:

```
if -1 <= x <= 1: y = 1
```

Задача 5

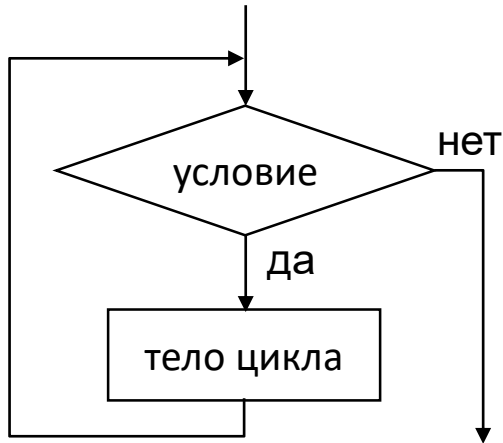
Вывести на экран название времени года по номеру месяца.

```
# Времена года (каскадное ветвление)
m = int(input("Введите номер месяца: "))
if m==1 or m==2 or m==12: print("Это зима")
elif m>=3 and m<=5: print("Это весна")
elif m>=6 and m<=8: print("Это лето")
elif m>=9 and m<=11: print("Это осень")
else: print("Неверный номер месяца!")
```

Операторы цикла

Цикл с предусловием

(с заданным условием продолжения работы, цикл «ПОКА»)



```
while <условие>:  
    <блок_операторов>
```

Выполнение *тела цикла* повторяется, **пока** условие **истинно**.

Если условие **ложно**, то управление передается следующему после цикла оператору.

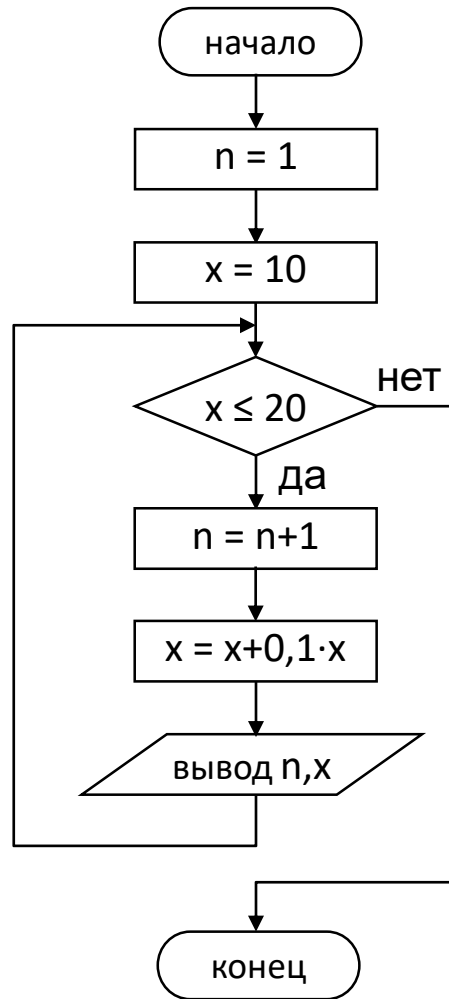
Если условие изначально ложно, тело цикла не выполнится ни разу.

Если условие никогда не станет ложным, то программа «зациклится».

Используется в основном тогда, когда *количество повторов заранее неизвестно*.

Задача 1

Лыжник в первый день тренировок пробежал 10 км. Каждый следующий день он увеличивал пройденное расстояние на 10% от пройденного в предыдущий день. В какой день он пробежит больше 20 км?

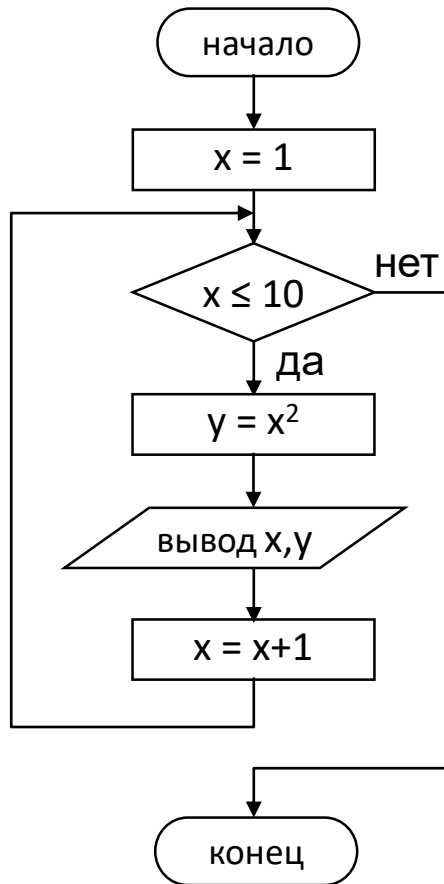


```
# Лыжная тренировка
n = 1                      # день номер 1
x = 10                     # расстояние в 1 день
while x <= 20:              # пока x <= 20 повторять:
    n = n + 1               # номер следующего дня
    x = x + 0.1 * x         # расстояние в следующий день
    print("{:3}".format(n), "{:6.1f}".format(x))
```

2	11.0
3	12.1
4	13.3
5	14.6
6	16.1
7	17.7
8	19.5
9	21.4

Задача 2

Получить таблицу значений функции $y=x^2$ для $x=1; 2; 3; \dots; 10$.



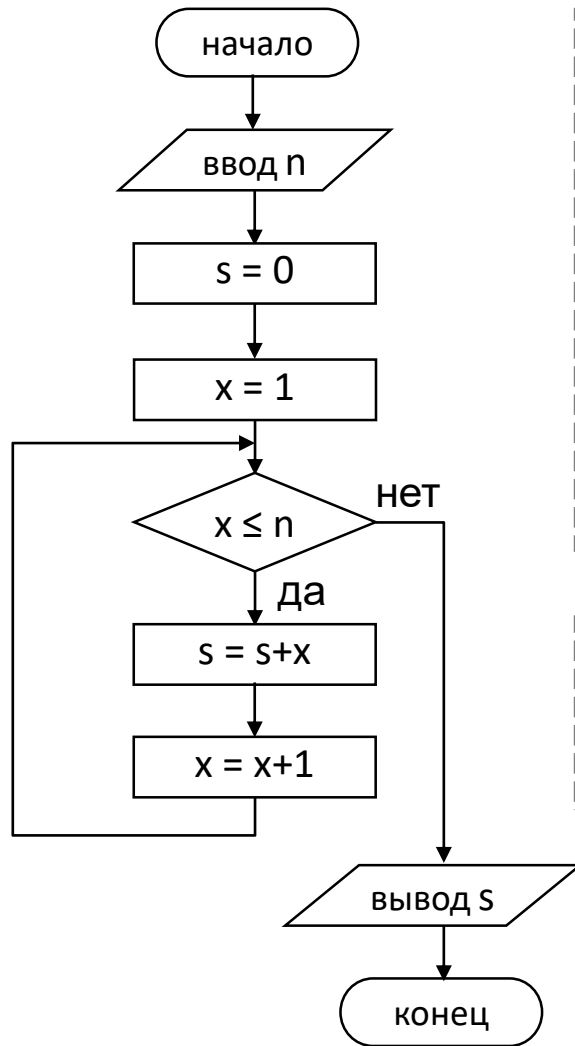
```
# Таблица квадратов
print("Квадраты чисел:")
x = 1 # начальное значение x
while x <= 10: # пока x <= 10 повторять:
    y = x**2 # вычисление функции
    print("{:3}".format(x), "{:5}".format(y))
    x = x + 1 # следующее значение x
```

Квадраты чисел:

1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

Задача 3а

Вычислить значение суммы $s = 1+2+3+\dots+n$ для заданного n .

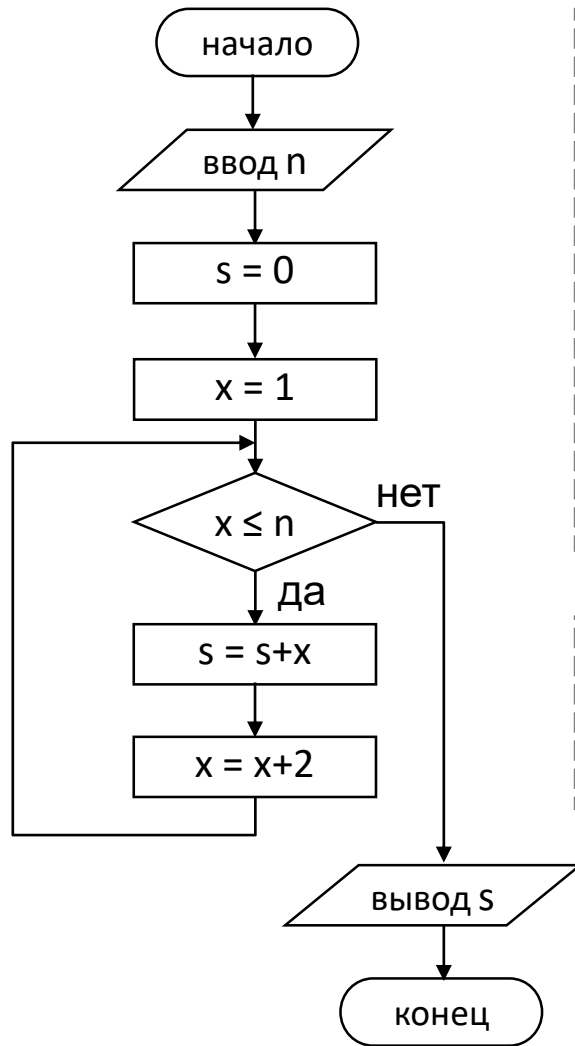


```
# Сумма натуральных чисел
print("s=1+2+3+...+n")
n = int(input("Введите n: "))
s = 0          # начальное значение суммы
x = 1          # начальное знач. слагаемого
while x <= n:  # пока x <= n повторять:
    s = s + x  # добавить к сумме слагаемое
    x = x + 1  # следующее знач. слагаемого
print("s=", s)
```

```
s=1+2+3+...+n
Введите n: 20
s= 210
```

Задача 36

Вычислить значение суммы $s = 1+3+5+7+\dots+n$ для заданного нечётного n .



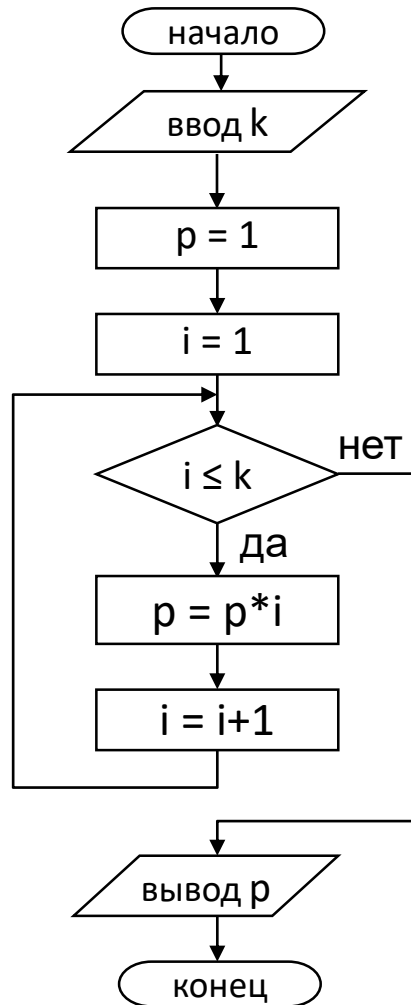
```
# Сумма нечетных натуральных чисел
print("s=1+3+5+...+n")
n = int(input("Ведите n: "))
s = 0          # начальное значение суммы
x = 1          # начальное знач. слагаемого
while x <= n:  # пока x <= n повторять:
    s = s + x  # добавить к сумме слагаемое
    x = x + 2  # следующее знач. слагаемого
print("s=", s)
```

```
s=1+3+5+...+n
Введите n: 19
s= 100
```

Задача 4

Вычислить факториал числа k .

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



```
# Факториал натурального числа
k = int(input("Введите k: "))
p = 1          # начальное знач. произвед.
i = 1          # начальное знач. множителя
while i <= k:   # пока i <= k повторять:
    p = p * i   # добавить к произведению
    i = i + 1   # следующее знач. множителя
print("Факториал числа равен", p)
```

```
Введите k: 5
Факториал числа равен 120
```

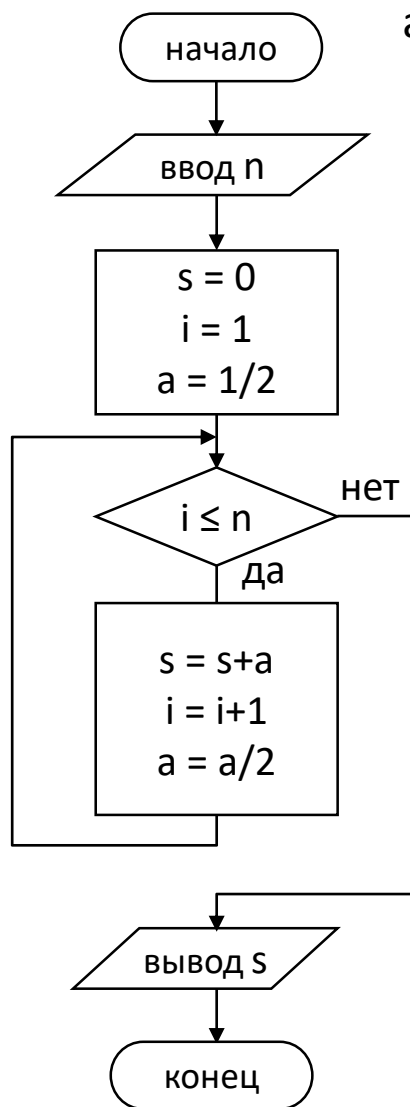
```
Введите k: 15
Факториал числа равен 1307674368000
```

Задача 5

Вычислить сумму ряда

$$S = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \dots (n \text{ слагаемых}).$$

$$a_1 = 1/2; a_2 = 1/4 = a_1/2; a_3 = 1/8 = a_2/2; \dots; a_i = a_{i-1}/2$$

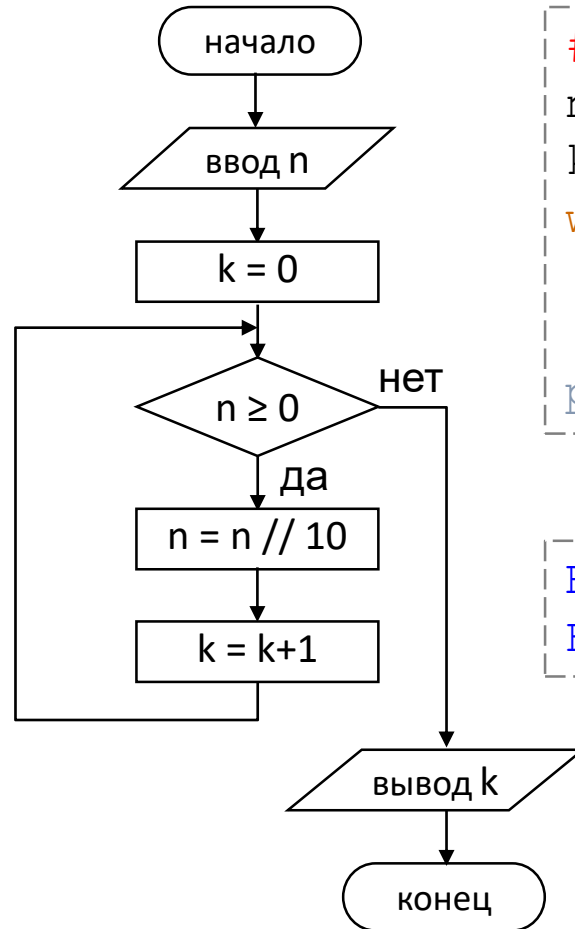


```
# Сумма ряда
print("S = 1/2+1/4+1/8+1/16+ ... ")
n = int(input("Введите n: "))
s = 0 # начальное значение суммы
i = 1 # номер первого слагаемого
a = 1/2 # первое слагаемое
while i <= n: # пока i <= n повторять:
    s = s + a # добавить к сумме слагаемое
    i = i + 1 # следующий номер слагаемого
    a = a / 2 # следующее слагаемое
print("S =", s)
```

```
S = 1/2+1/4+1/8+1/16+ ...
Введите n: 4
S = 0.9375
```

Задача 6

Подсчитать количество цифр в записи целого положительного числа.



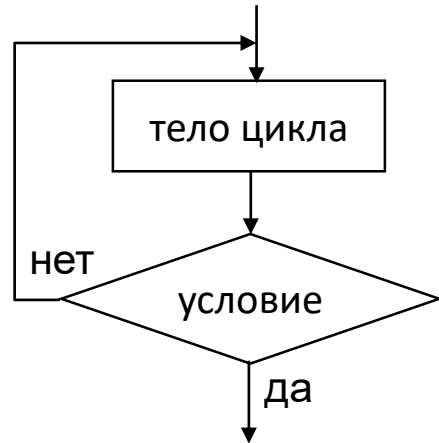
```
# Количество цифр целого числа
n = int(input("Введите число: "))
k = 0          # начальное значение счетчика
while n > 0:    # пока число > 0 повторять:
    n = n // 10 # отбрасываем последнюю цифру
    k = k + 1   # увеличиваем счетчик
print("Количество цифр в числе", k)
```

```
Введите число: 2020
Количество цифр в числе 4
```

Операторы цикла

Цикл с постусловием

(с заданным условием окончания работы, цикл «ДО»)



```
while True:
    <операторы>
    if <условие>: break
```

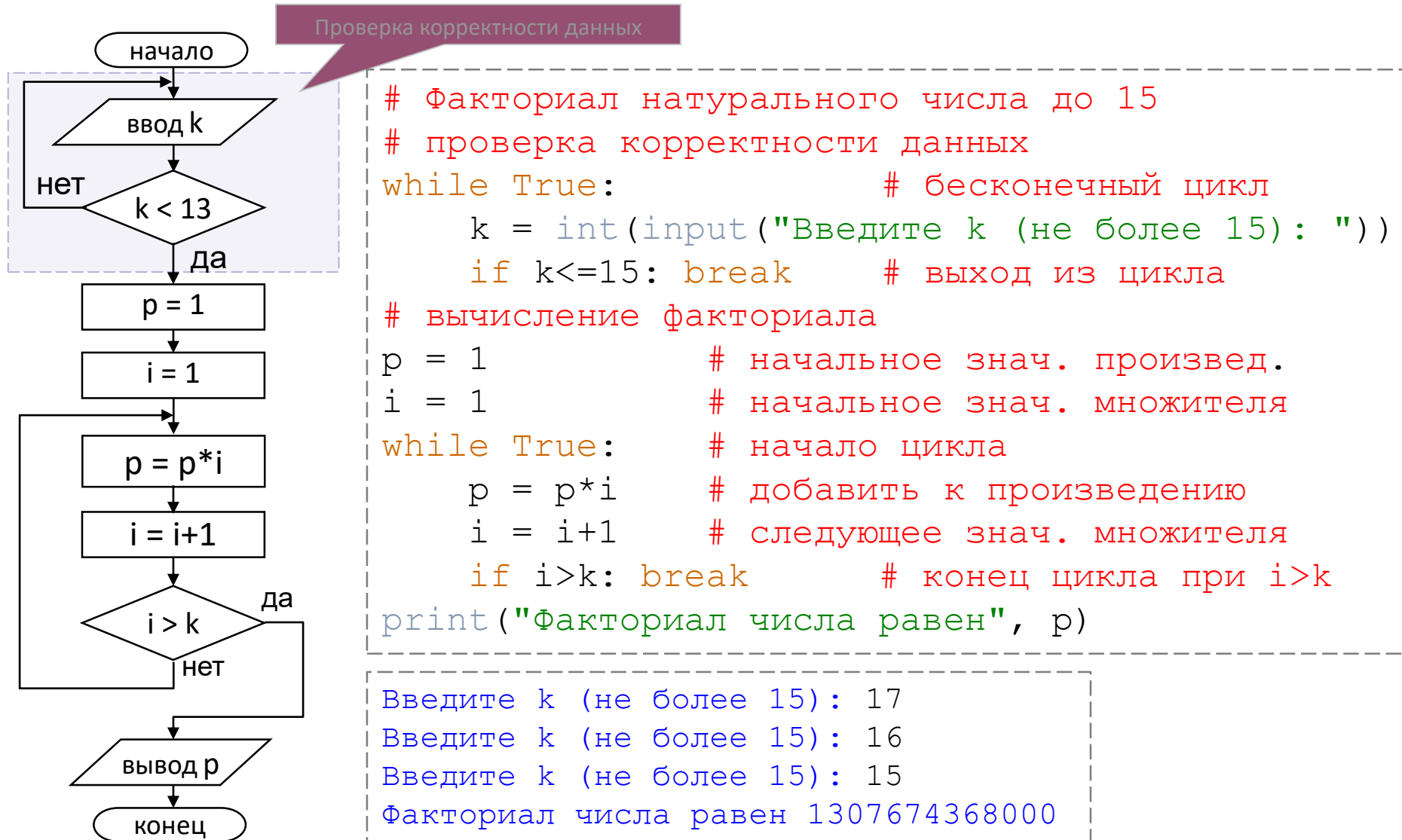
В языке Python нет оператора цикла с постусловием, но его можно организовать с помощью оператора **while** («пока») с условием **True** («истина»). Такой цикл будет выполняться бесконечно. Выход из цикла произойдет при истинности условия в операторе ветвления с помощью специального оператора **break** («прервать»).

Используется в тех случаях, когда требуется, чтобы *тело цикла выполнилось хотя бы один раз*.

Задача

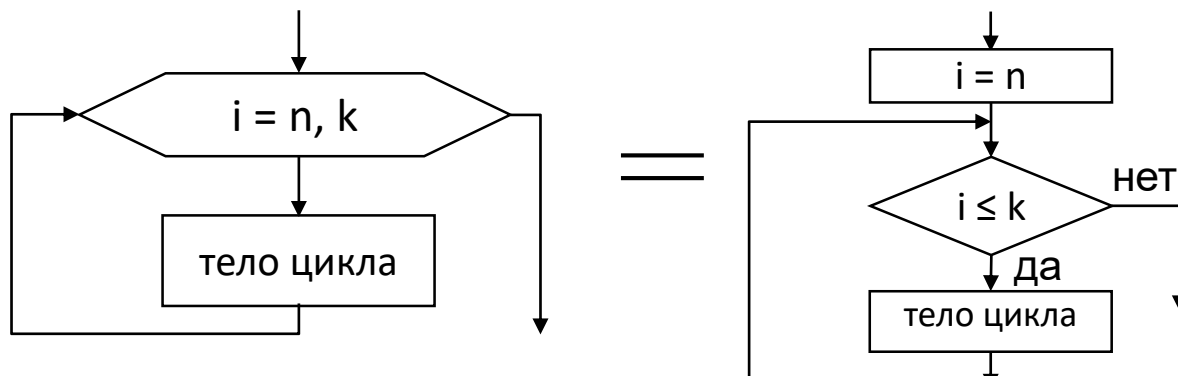
Вычислить факториал числа k (при k не более 15).

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



Операторы цикла

*Цикл с параметром
(с заданным числом повторений, цикл «ДЛЯ»)*



```
for <параметр> in range (<диапазон>) :  
    <операторы_тела_цикла>
```

Тело цикла повторяется фиксированное число раз **для** каждого значения параметра. Параметр – переменная целого типа. Функция **range** («диапазон») задаёт количество повторов тела цикла и содержит от одного до трёх чисел.

- Одно число (**k**) – параметр цикла изменяется от 0 до k-1 с шагом 1.
- Два числа (**n, k**) – параметр цикла изменяется от n до k-1 с шагом 1.
- Три числа (**n, k, s**) – параметр цикла изменяется от n до k-1 с шагом s.

Возможно изменение параметра от большего значения к меньшему. В этом случае **n** должно быть больше **k**, а **s** – отрицательное.

Примеры записи оператора цикла с параметром

```
# для i в диапазоне от 0 до 9
for i in range(10):
    print(i, end=" ")
print("Всё!")
```

0 1 2 3 4 5 6 7 8 9 Всё!

```
# для i в диапазоне от 1 до 9
for i in range(1,10):
    print(i, end=" ")
print("Всё!")
```

1 2 3 4 5 6 7 8 9 Всё!

```
# для i в диапазоне от 1 до 9 с шагом 2
for i in range(1,10,2):
    print(i, end=" ")
print("Всё!")
```

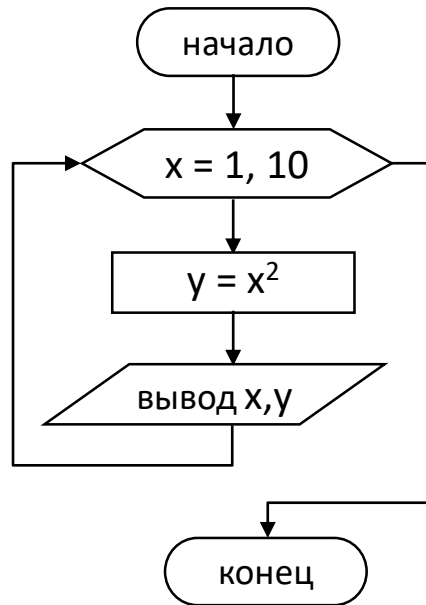
1 3 5 7 9 Всё!

```
# для i в диапазоне от 9 до 1 с шагом -2
for i in range(9,0,-2):
    print(i, end=" ")
print("Всё!")
```

9 7 5 3 1 Всё!

Задача 1

Получить таблицу значений функции $y=x^2$ для $x=1; 2; 3; \dots; 10$.



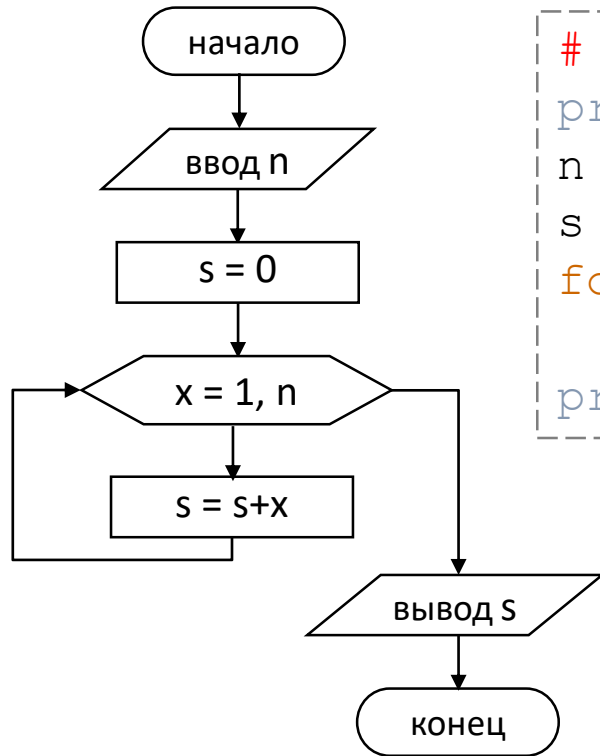
```
# Таблица квадратов
print("Квадраты чисел:")
for x in range(1, 11): # для i от 1 до 10
    y = x**2           # вычисление функции
    print("{:3}".format(x), "{:5}".format(y))
```

Квадраты чисел:

1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

Задача 2

Вычислить значение суммы $s = 1+2+3+\dots+n$ для заданного n .



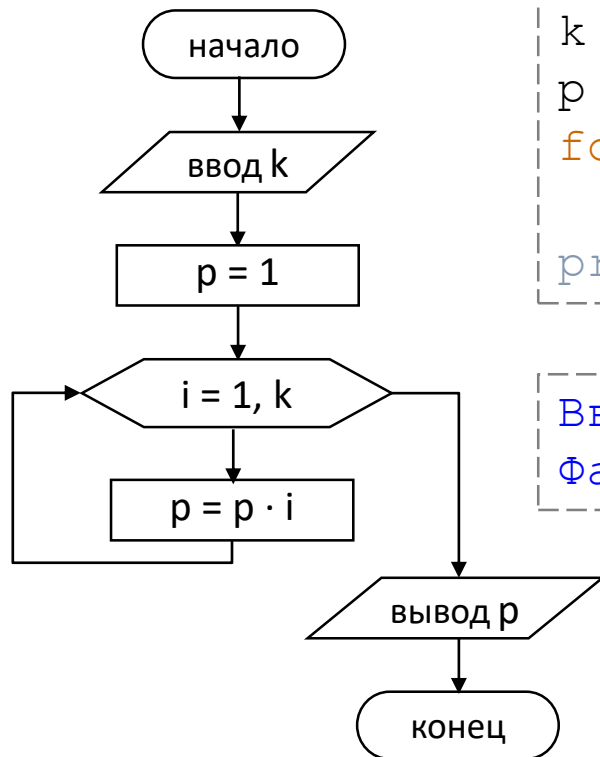
```
# Сумма натуральных чисел
print("s=1+2+3+...+n")
n = int(input("Ведите n: "))
s = 0 # начальное значение суммы
for x in range(1, n+1): # для i от 1 до n
    s = s+x # добавить к сумме слагаемое
print("s=", s)
```

```
s=1+2+3+...+n
Введите n: 20
s= 210
```

Задача 3

Вычислить факториал числа k .

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



```
# Факториал натурального числа
k = int(input("Введите k: "))
p = 1 # начальное знач. произвед.
for i in range(1, k+1): # для i от 1 до k
    p = p*i # добавить к произведению
print("Факториал числа равен", p)
```

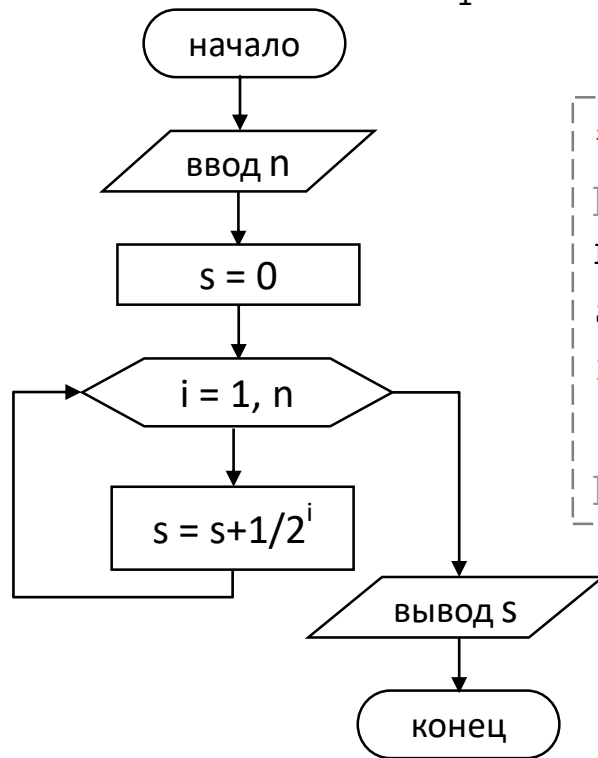
```
Введите k: 5
Факториал числа равен 120
```

Задача 4

Вычислить сумму ряда

$$S = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \dots + \frac{1}{2^n} \text{ (n слагаемых).}$$

$$a_1 = 1/2^1; a_2 = 1/2^2; a_3 = 1/2^3; \dots; a_i = 1/2^i$$

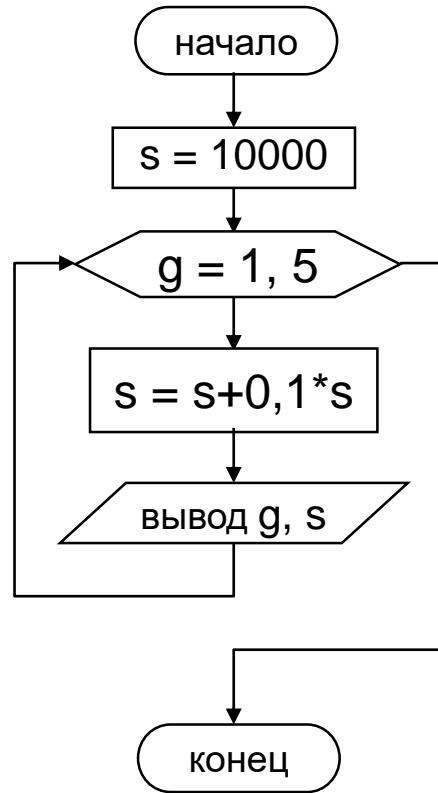


```
# Сумма ряда
print("S = 1/2+1/4+1/8+1/16+ ... ")
n = int(input("Введите n: "))
s = 0 # нач. знач. суммы
for i in range(1, n+1): # для i от 1 до n
    s = s + 1/2**i # добавить к сумме
print("S =", s)
```

```
S = 1/2+1/4+1/8+1/16+ ...
Введите n: 4
S = 0.9375
```

Задача 5

Банк принимает вклады под 10% годовых. На счет положена сумма 10000 рублей. Какая сумма будет на счету через 5 лет?



```
# Вклад в банке
s = 10000                                # начальный вклад
print("Начальная сумма", s)
for g in range(1, 6):                    # для g от 1 до 5
    s = s + 0.1*s                         # добавление 10%
    print(g, "год сумма", s)
```

```
Начальная сумма 10000
1 год сумма 11000.0
2 год сумма 12100.0
3 год сумма 13310.0
4 год сумма 14641.0
5 год сумма 16105.1
```