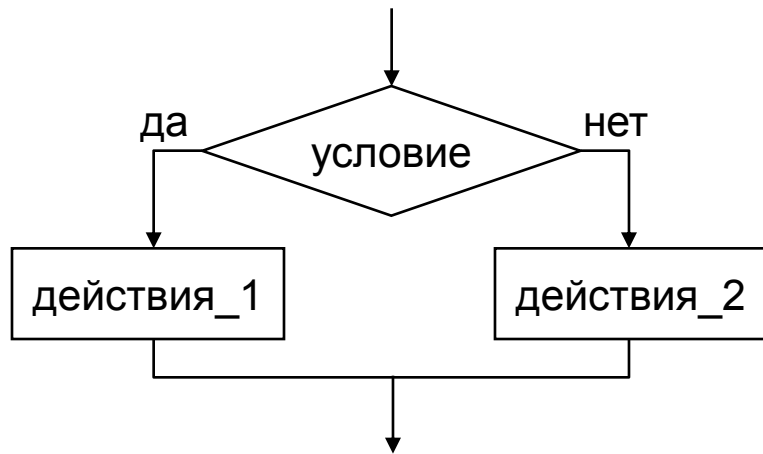




Операторы ветвления

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**.*

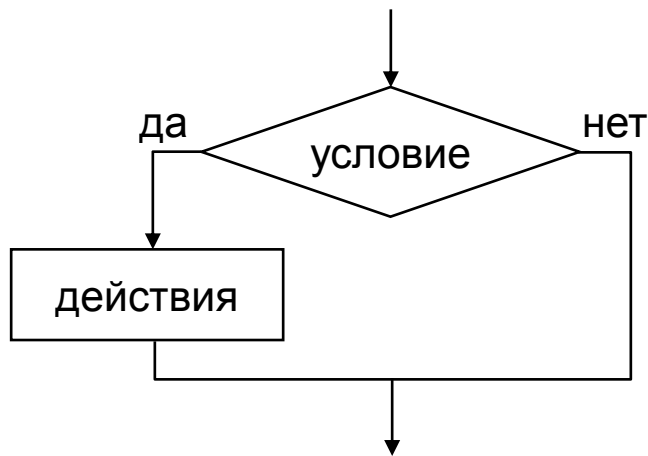
Если в блоке всего один оператор, то допустима такая форма записи:

```
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>
```

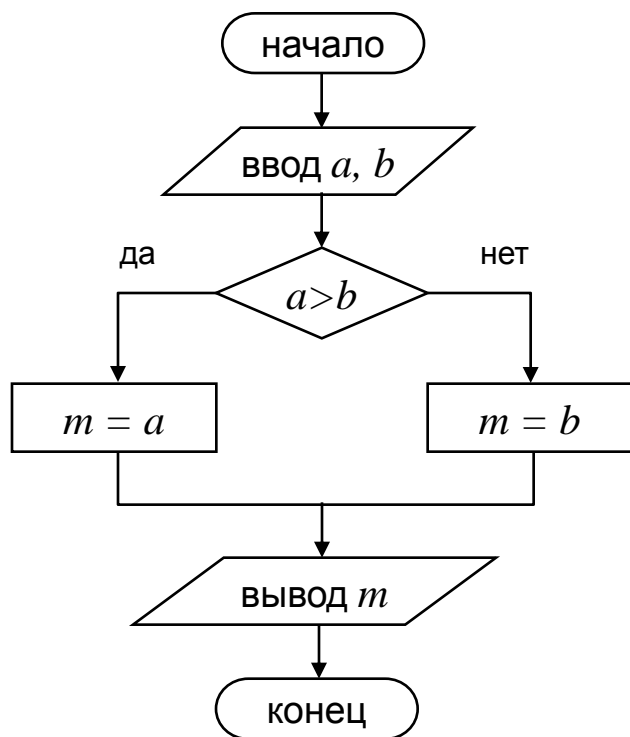
```
if <условие_1>:  
    <блок_1>  
elif <условие_2>:  
    <блок_2>  
...  
else:  
    <блок_N>
```

Задача 1

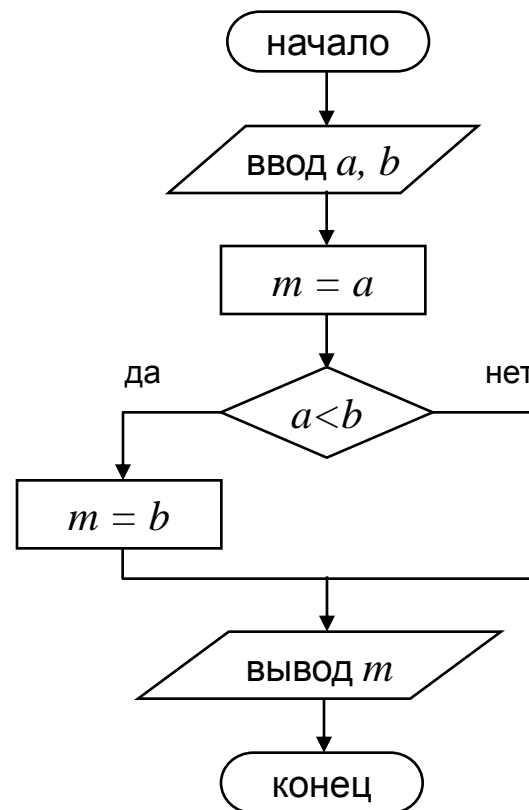
Определить большее из двух значений переменных.



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 = b
else: m = a
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="))
```

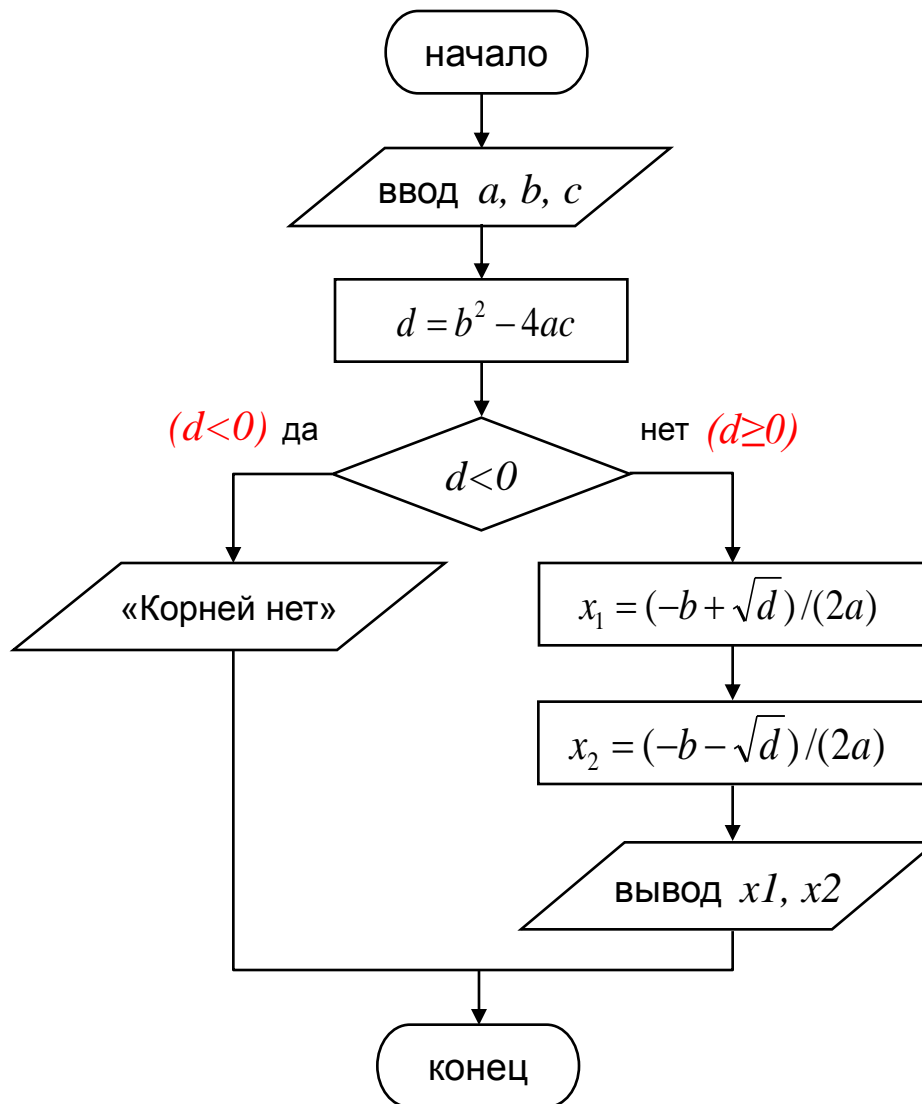
```
Введите два числа:
a=6
b=5
Большее число 6
```

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:

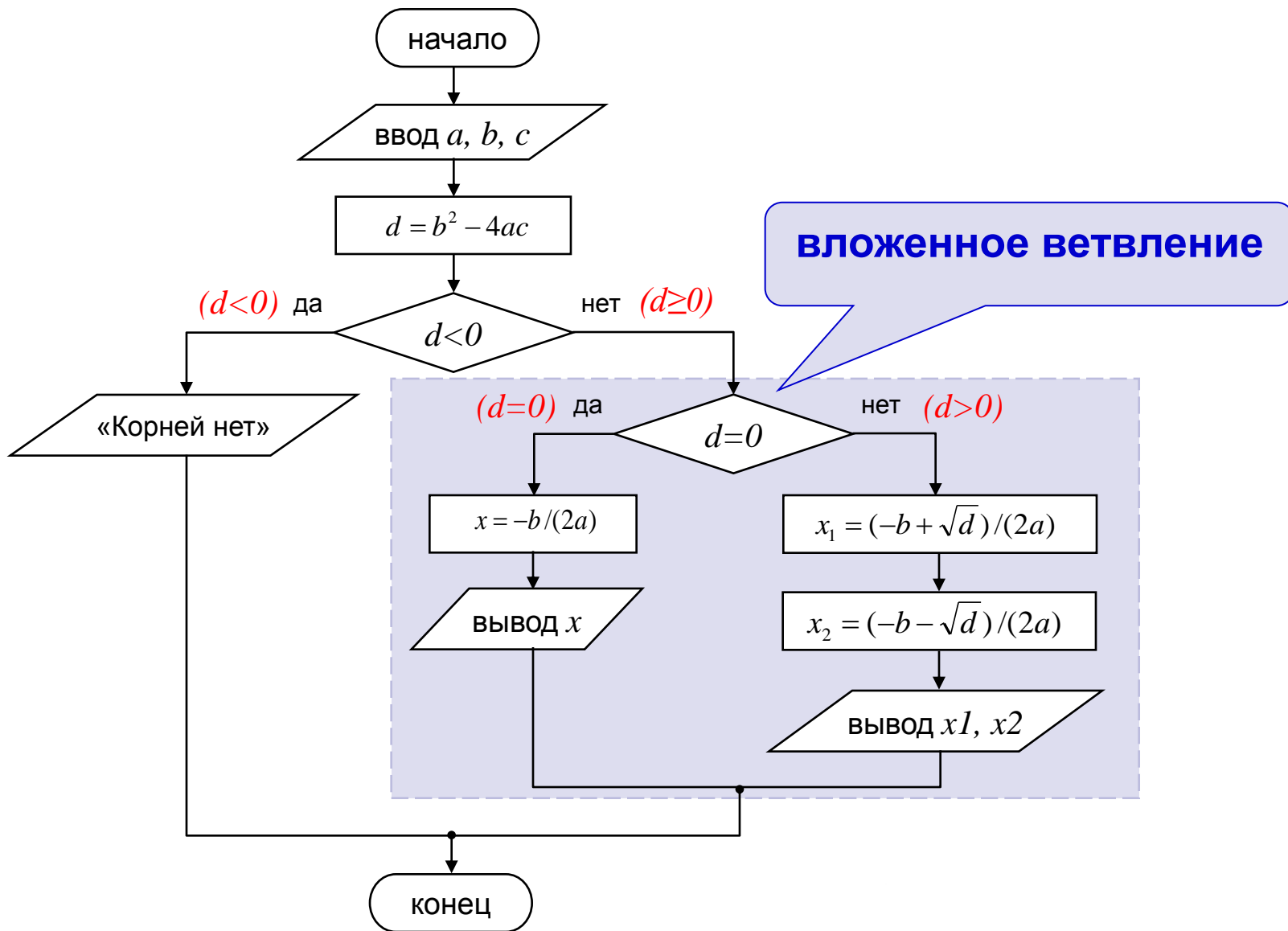
```
a= 1
b= 3
c= 2
x1=  -1.00
x2=  -2.00
```

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))
```

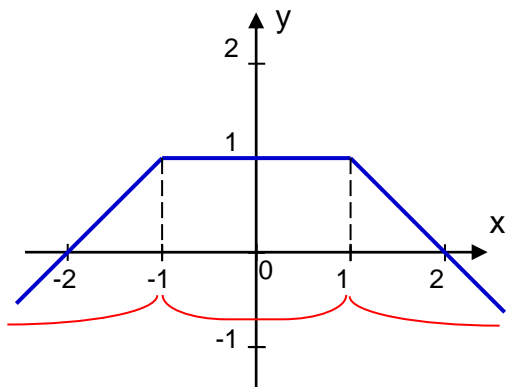
Введите коэффициенты a, b, c:
a= 4
b= 1
c= 4
Корней нет!

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

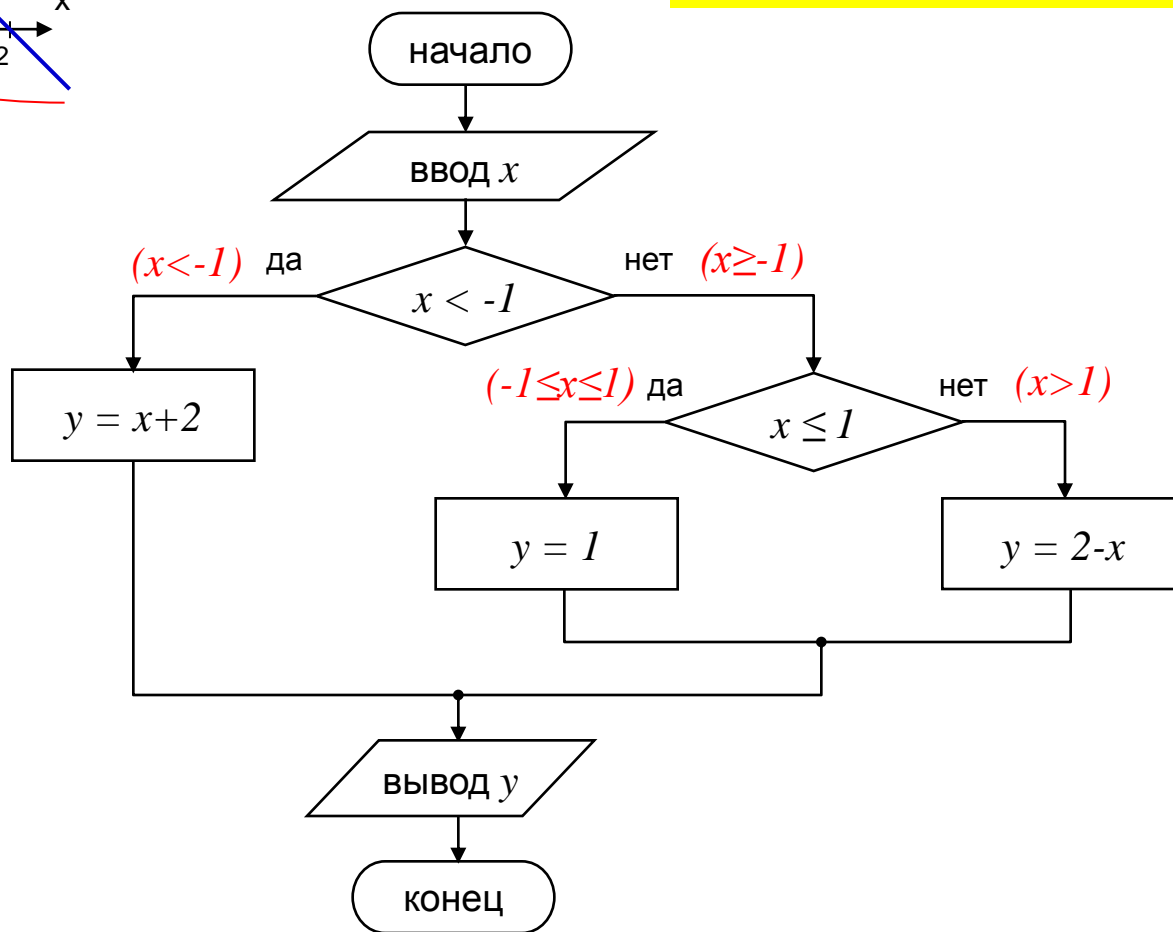
Введите коэффициенты a, b, c:
a= 1
b= 3
c= 2
x1= -1.00
x2= -2.00

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

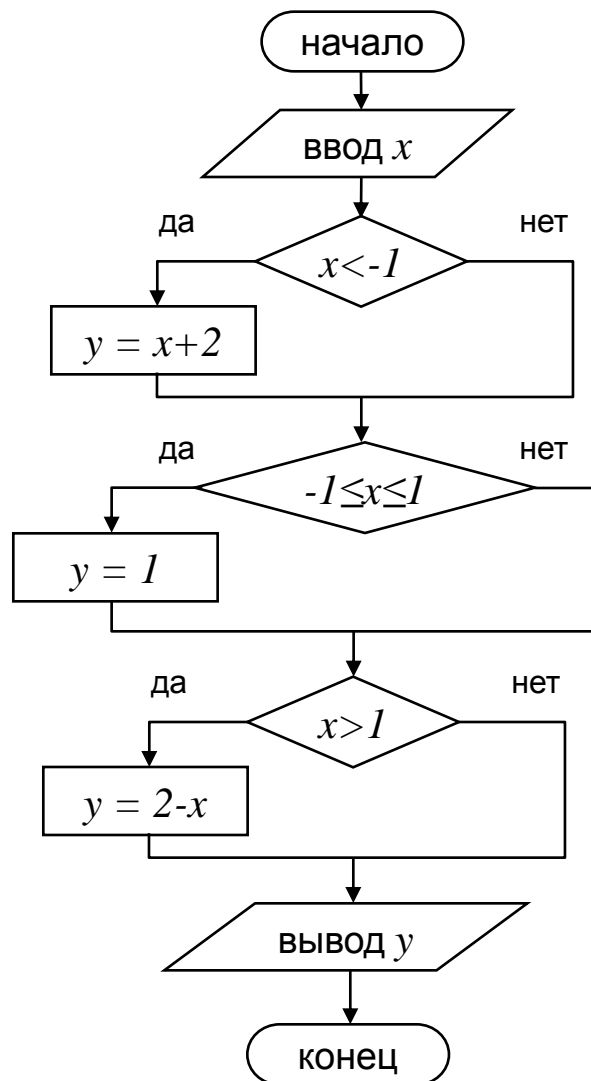
Введите x: 2
y= 0.0

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
```



Источник <https://ppt-online.org/862442>