



**ИНФОРМАТИКА**

**8**

класс

# ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЯЮЩИХСЯ АЛГОРИТМОВ

НАЧАЛА ПРОГРАММИРОВАНИЯ  
НА ЯЗЫКЕ PYTHON

# КЛЮЧЕВЫЕ СЛОВА

- ◆ разветвляющиеся алгоритмы
- ◆ условный оператор
- ◆ неполный условный оператор
- ◆ каскадное ветвление

# РАЗВЕТВЛЯЮЩИЕСЯ АЛГОРИТМЫ

**Алгоритмы с ветвлениями** - алгоритмы, в которых в зависимости от результата проверки условия («да» или «нет») предусмотрен выбор одной из двух последовательностей действий (ветвей).



# ОБЩИЙ ВИД УСЛОВНОГО ОПЕРАТОРА

## Полная форма:

```
if <условие>:  
    <группа операторов 1>  
else:  
    <группа операторов 2>
```

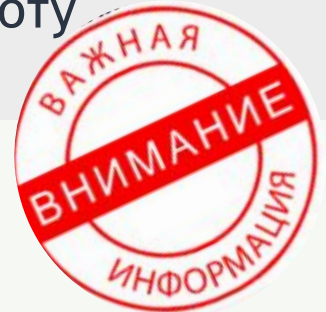
## Неполная форма:

```
if <условие>:  
    <операторы>
```

Двойные неравенства  
 $A < B < C$

*Обратите внимание!* Слова `if` и `else` начинаются на одном уровне, а все команды внутренних блоков сдвинуты относительно этого уровня вправо на одно и то же расстояние.

В языке Python сдвиги операторов относительно левой границы (отступы) влияют на работу программы.



Условия - логические выражения

Простые

Сложные

Операции  
отношений  
 $\geq, \leq, !=, ==$

Логические  
операции  
`and, or, not`



# ЧИТАЕМ ПРОГРАММНЫЙ КОД

```
x = int(input('Введите x: '))  
if x % 2 == 0:  
    print(x, ' - чётное число')  
else:  
    print(x, ' - нечётное число')
```

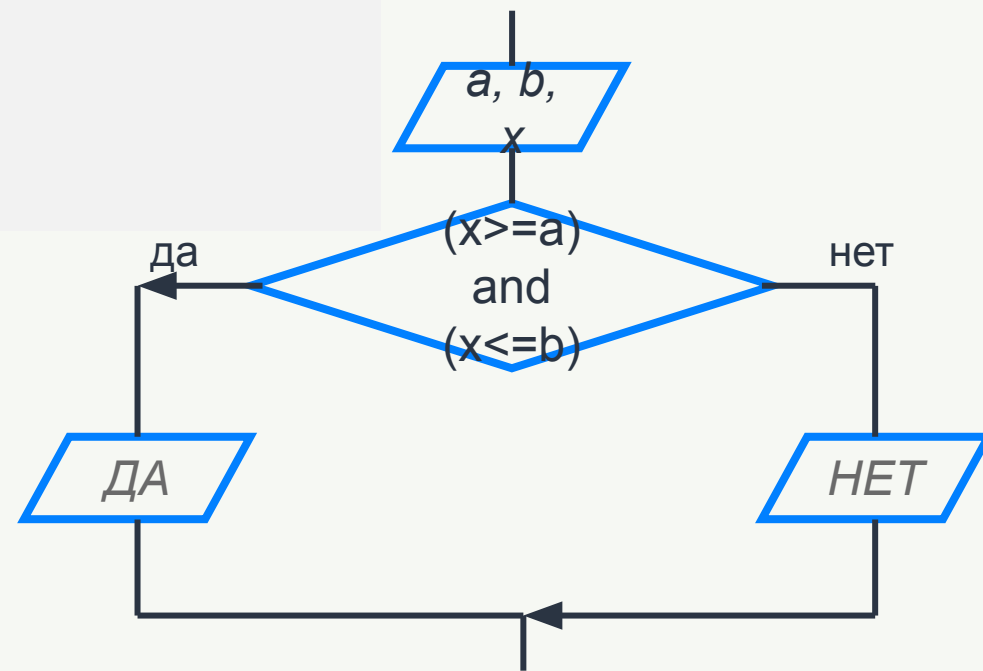
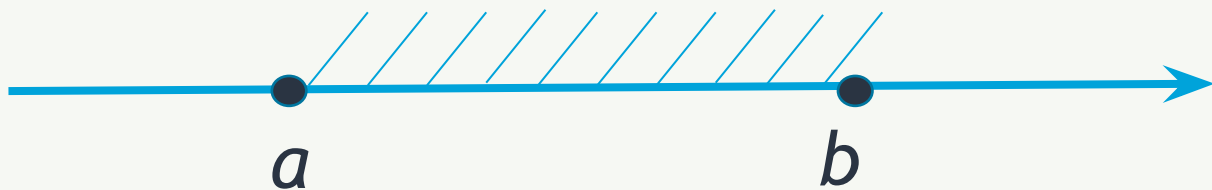


Измените программу так, чтобы она определяла кратность исходного числа пяти.



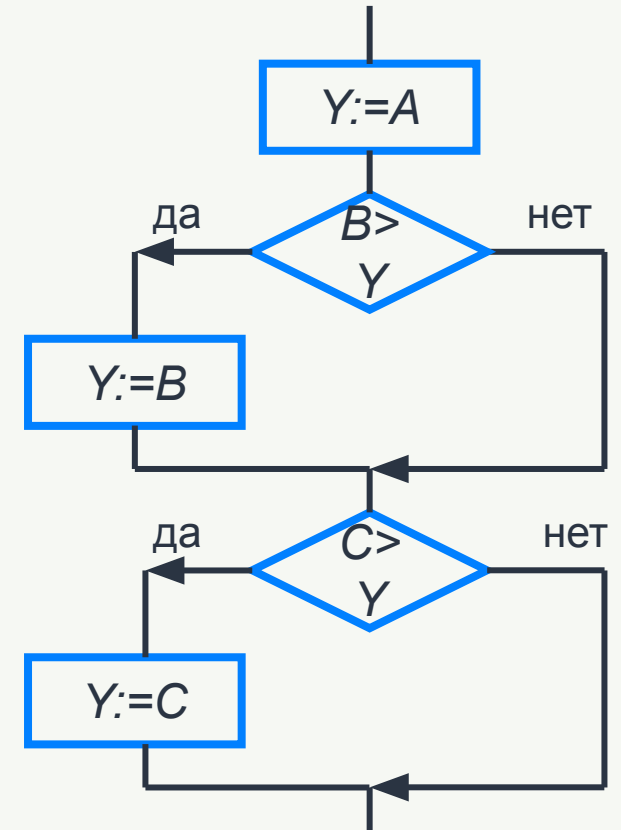
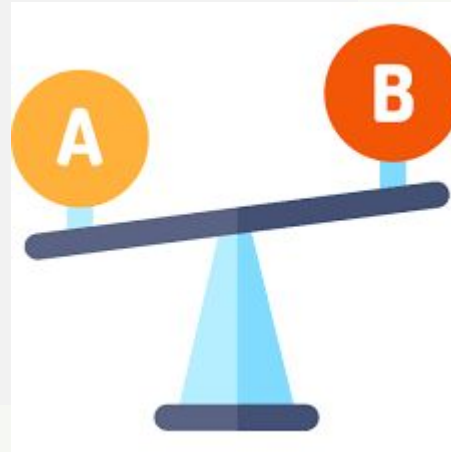
# ЧИТАЕМ ПРОГРАММНЫЙ КОД

```
a = int (input ('Введите a: '))  
b = int (input ('Введите b: '))  
x = int (input ('Введите x: '))  
if x >= a and x <= b:  
    print ('Да')  
else:  
    print ('Нет')
```



# ЧИТАЕМ ПРОГРАММНЫЙ КОД

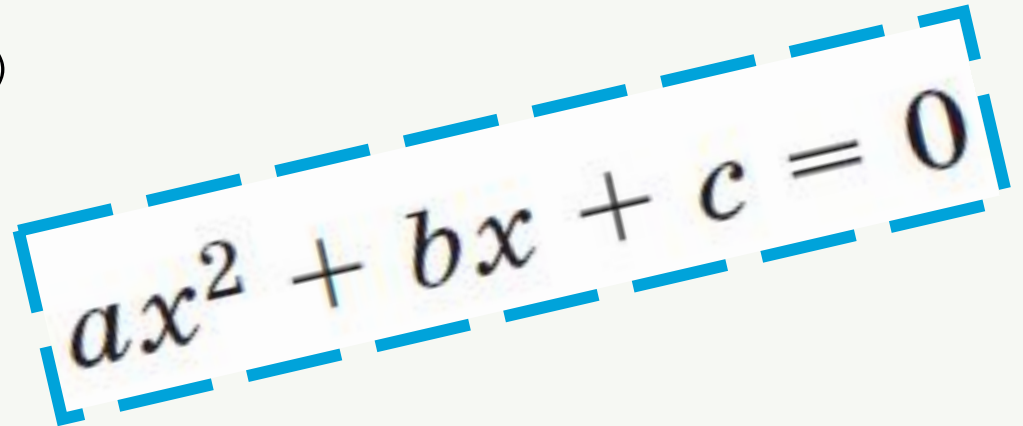
```
a = int(input('Введите a: '))
b = int(input('Введите b: '))
c = int(input('Введите c: '))
y = a
if b > y:
    y = b
if c > y:
    y = c
print('y=', y)
```



Измените программу так, чтобы она определяла наибольшее из четырёх чисел.

# ЧИТАЕМ ПРОГРАММНЫЙ КОД

```
from math import *
print ('Введите коэффициенты a, b, c>>')
a = float (input ('a='))
b = float (input ('b='))
c = float (input ('c='))
d = b * b - 4 * a * c
if d<0:
    print ('Корней нет')
if d == 0:
    x = - b/2/ a
    print ('Корень уравнения x=', "{:6.4f}". format(x))
if d>0:
    x1 = (-b+sqrt(d))/2/ a
    x2 = (-b - sqrt(d))/2/ a
    print ('Корни уравнения:')
    print ('x1=', "{:6.4f}". format(x1))
    print ('x2=', "{:6.4f}". format(x2))
```


$$ax^2 + bx + c = 0$$



# ВЛОЖЕННЫЕ ВЕТВЛЕНИЯ

```
if <условие 1> :  
    <группа операторов 1>  
else:  
    if <условие 2> :  
        <группа операторов 2>  
    else:  
        <группа операторов 3>
```

```
if <условие 1> :  
    if <условие 2> :  
        <группа операторов 1>  
    else:  
        <группа операторов 2>  
else:  
    <группа операторов 3>
```

Внутри условного оператора могут находиться любые операторы, в том числе и другие условные операторы.

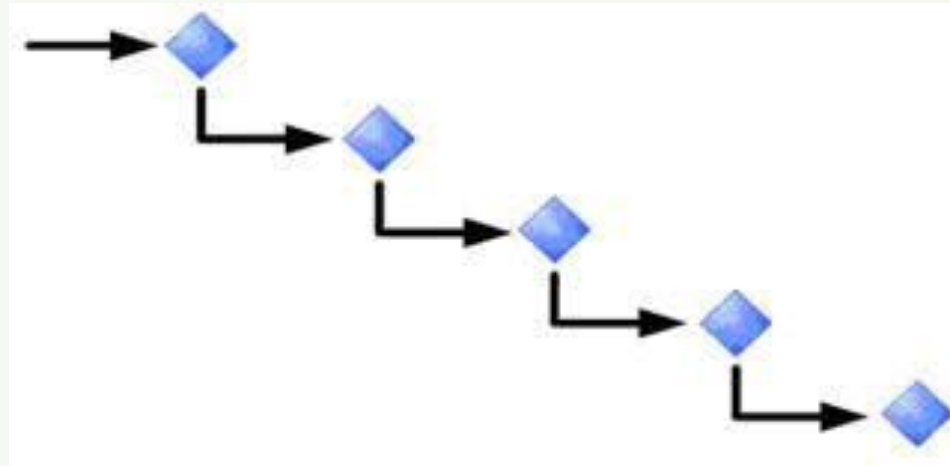


# КАСКАДНОЕ ВЕТВЛЕНИЕ

Если после **else** сразу следует еще один оператор **if**, можно использовать каскадное ветвление со служебным словом **elif** (сокращение от **else-if**):

```
if <условие1>:  
    <оператор 1>  
elif <условие2>:  
    <оператор 2>  
    ...  
elif <условие n>:  
    <оператор n>  
else :  
    <оператор m>
```

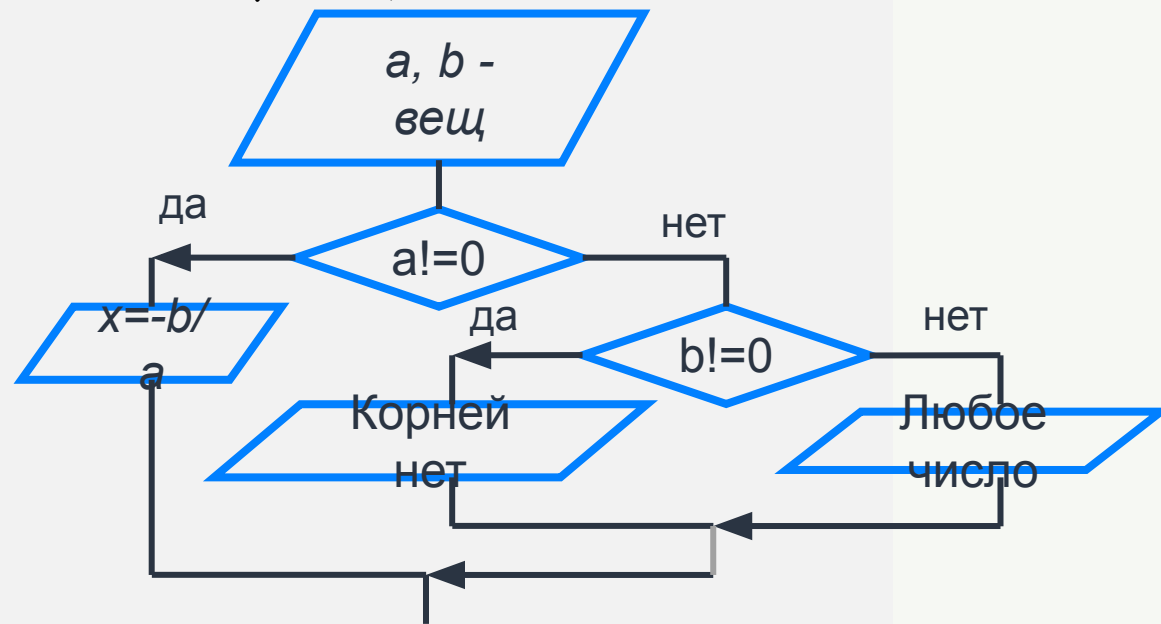
Каскадное ветвление позволяет выбрать один из нескольких (а не только из двух) вариантов.



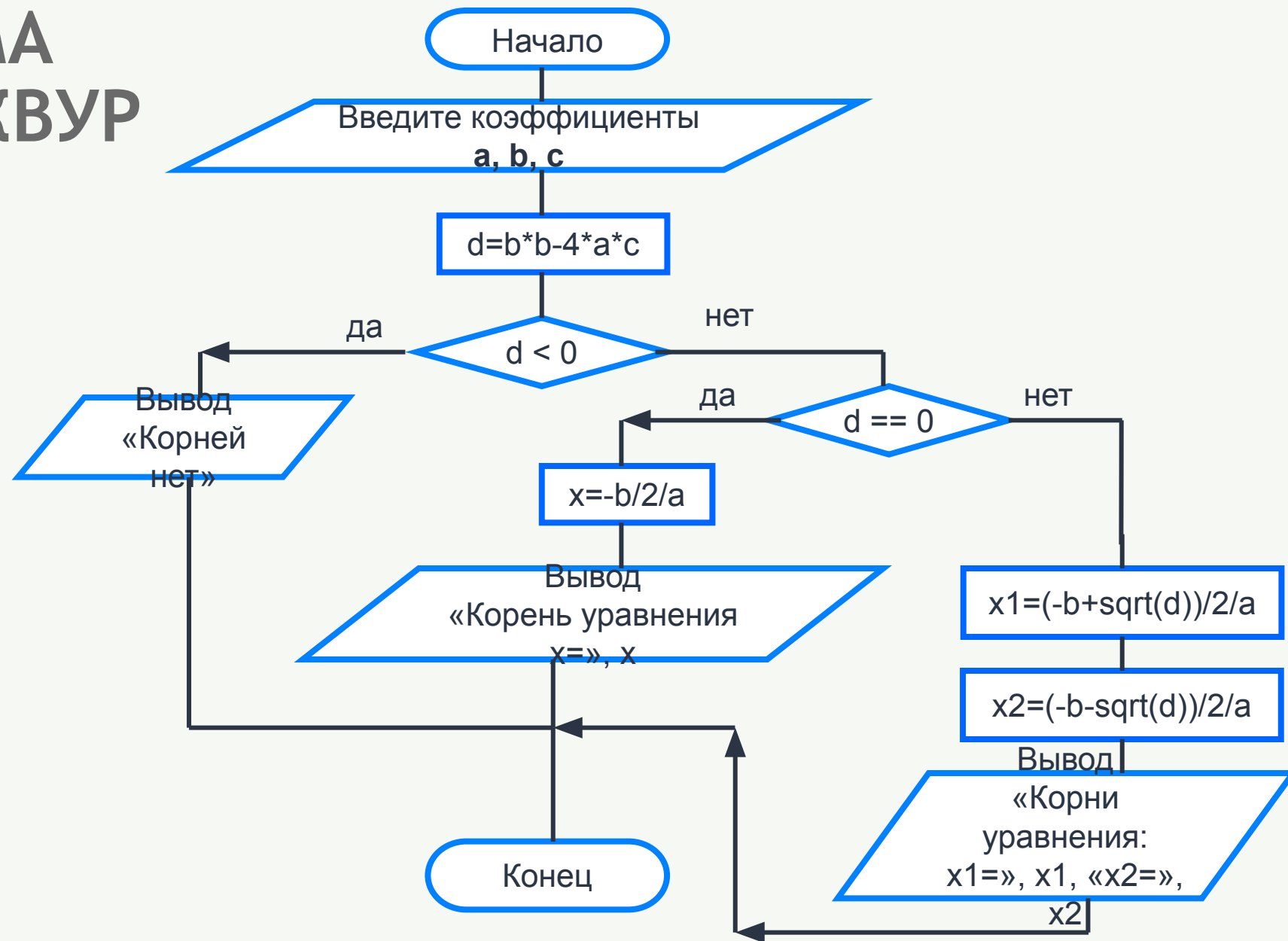
# РЕШЕНИЕ ЛИНЕЙНОГО УРАВНЕНИЯ

```
a = float( input ( 'Введите коэффициент a>>' ) )
b = float( input ( 'Введите коэффициент b>>' ) )
if a != 0:
    x = -b/ a
    print ( 'Корень уравнения x=', x )
elif b != 0:
    print ( 'Корней нет' )
else:
    print ( ' x-любое число' )
```

$$a \cdot x + b = 0$$



# БЛОК-СХЕМА РЕШЕНИЯ КВУР



```
from math import *
print ('Введите коэффициенты a, b, c>>')
a = float (input ('a='))
b = float (input ('b='))
c = float (input ('c='))
d = b * b - 4 * a * c
if d<0:
    print ('Корней нет')
elif d == 0:
    x = - b/2/ a
    print ('Корень уравнения x=', "{:6.4f}". format(x))
else:
    x1 = (-b+sqrt(d))/2/ a
    x2 = (-b - sqrt(d))/2/ a
    print ('Корни уравнения:')
    print ('x1=', "{:6.4f}". format(x1))
    print ('x2=', "{:6.4f}". format(x2))
```

Дана программа на языке Python:

```
s = int(input())
t = int(input())
if s // 2 == t :
    print('YES')
else:
    print ('NO')
```

**Ответ: 5 раз**

Было проведено 9 запусков программы, при которых в качестве значений переменных  $s$  и  $k$  вводились следующие пары чисел:

(1, 1); (8, 4); (14, 10); (20, 1); (7, 3); (10, 5); (10, 2); (4, 1); (1, 0).

Сколько было запусков, при которых программа напечатала «NO»?

$s$	$t$	$s \text{ div } 2 = t$	Вывод
1	1	0	NO
8	4	4	YES
14	10	7	NO
20	1	10	NO
7	3	3	YES
10	5	5	YES
10	2	5	NO
4	1	2	NO
1	0	0	YES

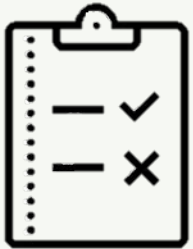
При записи на языке Python разветвляющихся алгоритмов используют условный оператор, позволяющий выбрать один из двух вариантов действий в зависимости от выполнения некоторого условия:

```
if <условие>:  
    <группа операторов 1>  
else:  
    <группа операторов 2>
```

Для записи неполных ветвлений используется неполный условный оператор:

```
if <условие>:  
    <операторы>
```

В обеих частях условного оператора можно использовать любые операторы, в том числе и другие (вложенные) условные операторы.



# ВОПРОСЫ И ЗАДАНИЯ

Является ли условным оператором последовательность символов?

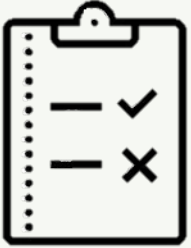
а) **if**  $x < y$ :  $x = 0$  **else**: `input (y)`

б) **if**  $x \geq y$ :  $x = 0$ ;  $y := 0$   
**else**: `print (z)`

в) **if**  $x < y < z$ :  $a = a + 1$







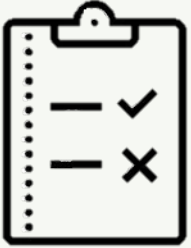
# ВОПРОСЫ И ЗАДАНИЯ

Дан условный оператор:

```
if a < 5: c = 1
else:
    if a > 5: c = 2
    else: c = 3
```

Какое значение имеет переменная  $a$ , если в результате выполнения условного оператора переменной  $c$  присваивается значение 3?





# ВОПРОСЫ И ЗАДАНИЯ

Используя полное ветвление, упростите следующий фрагмент программы:

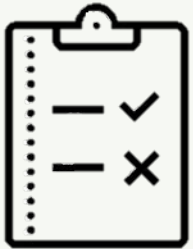
```
if a > b: c = 1
```

```
if a > b: d = 2
```

```
if a <= b: c = 3
```

```
if a <= b: d = 4
```



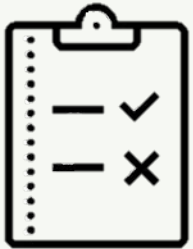


# ВОПРОСЫ И ЗАДАНИЯ

Напишите программу, которая производит обмен значений переменных  $x$  и  $y$ , если  $x$  больше  $y$ .

Пример входных данных	Пример выходных данных
$x \gg 5$ $y \gg 6$	$x = 5$ $y = 6$
$x \gg 6$ $y \gg 5$	$x = 5$ $y = 6$



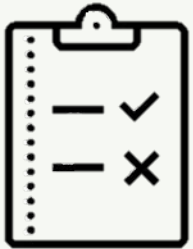


# ВОПРОСЫ И ЗАДАНИЯ

Дано трёхзначное число. Напишите программу, которая определяет, является ли число «перевёртышем», т. е. числом, десятичная запись которого читается одинаково слева направо и справа налево.

Пример входных данных	Пример выходных данных
123	Нет
121	Перевёртыш
222	Перевёртыш



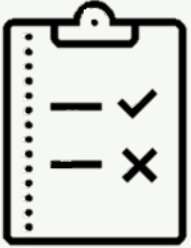


# ВОПРОСЫ И ЗАДАНИЯ

Даны три натуральных числа. Напишите программу, определяющую, существует ли треугольник с такими длинами сторон. Если такой треугольник существует, то определите его тип (равносторонний, равнобедренный, разносторонний).

Пример входных данных	Пример выходных данных
<code>a b c &gt;&gt; 1 2 1</code>	Не существует
<code>a b c &gt;&gt; 2 2 2</code>	Равносторонний
<code>a b c &gt;&gt; 20 20 30</code>	Равнобедренный
<code>a b c &gt;&gt; 3 4 5</code>	Разносторонний





# ВОПРОСЫ И ЗАДАНИЯ

Имеются данные о количестве полных лет трёх призёров спартакиады. Напишите программу, выбирающую и выводящую возраст самого младшего призёра.



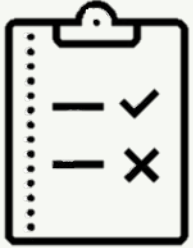


# ВОПРОСЫ И ЗАДАНИЯ

Поле шахматной доски определяется парой натуральных чисел, каждое из которых не превосходит 8. Напишите программу, которая по введённым координатам двух полей  $(k, l)$  и  $(m, n)$  определяет, являются ли эти поля полями одного цвета.

Пример входных данных		Пример выходных данных	
Координаты 1 – го поля >>	2 2	Поля одного цвета	
Координаты 2 – го поля >>	3 3		
Координаты 1 – го поля >>	2 3	Поля разного цвета	
Координаты 2 – го поля >>	3 3		
Координаты 1 – го поля >>	2 7	Поля одного цвета	
Координаты 2 – го поля >>	5 4		





# ВОПРОСЫ И ЗАДАНИЯ

Напишите программу, в которой пользователю предлагается дополнить до 100 некоторое целое число  $a$  ( $a$  - случайное число, меньше 100).

Ответ пользователя проверяется и комментируется.





# ПРОГРАММИРОВАНИЕ ВЕТВЛЕНИЙ

## УСЛОВНЫЙ ОПЕРАТОР

### Полная форма

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

### Неполная форма

```
if <условие>: <оператор>
```

### Каскадное ветвление

```
if <условие1>: <оператор 1>  
elif <условие2>: <оператор 2>  
...  
elif <условие n>: <оператор n>  
else : <оператор m>
```