

Условный оператор позволяет выполнять действия в зависимости от того, выполнено условие или нет.

Конструкция «если»

Блок команд, который будет выполняться, выделяется отступами в 4 пробела. Записывается условный оператор следующим образом:

```
if <логическое выражение>:  
    <действие1>  
    <действие2>  
<действие3>
```

Внутри логической конструкции выполняются <действие1>, <действие2>, а <действие3> выполнится в любом случае, истинно или ложно условие.

Пример. Задача о нахождении модуля числа.

В этой программе с отступом записана только одна строка, $x = -x$. При необходимости выполнить несколько команд все они должны быть записаны с тем же отступом. Команда `print` записана без отступа, поэтому она будет выполняться в любом случае, независимо от того, было ли условие в `if` истинным или нет.

```
x = int(input())  
if x < 0:  
    x = -x  
print(x)
```

В дополнение к `if` можно использовать оператор `else`: (иначе). Блок команд, который следует после него, будет выполняться если условие было ложным. Например, ту же задачу о выводе модуля числа можно было решить, не меняя значения переменной `x`:

```
x = int(input())  
if x >= 0:  
    print(x)  
else:  
    print(-x)
```

Все команды, которые выполняются в блоке `else`, должны быть также записаны с отступом. `Else` должен следовать сразу за блоком команд `if`, без промежуточных команд, выполняемых безусловно. `Else` без соответствующего `if` а не имеет смысла.

Пример.

Напишите программу, которая считывает два целых числа А и В и выводит наибольшее значение из них. Числа — целые от 1 до 1000.

Формат ввода

Вводятся два числа.

Формат вывода

Выведите ответ на задачу.

Тест 1

Входные данные:

8

5

Вывод программы:

8

Тест 2

Входные данные:

5

8

Вывод программы:

8

Тест 3

Входные данные:

5

5

Вывод программы:

5

Решение:

```
a = int(input())
```

```
b = int(input())
```

```
if a > b:
```

```
    print(a)
```

```
else:
```

```
    print(b)
```

Вложенный условный оператор

Внутри блока команд могут находиться другие условные операторы. По заданному количеству глаз и ног нужно научиться отличать кошку, паука, морского гребешка и жучка. У морского гребешка бывает более сотни глаз, а у пауков их восемь. Также у пауков восемь ног, а у морского гребешка их нет совсем. У кошки четыре ноги, а у жучка – шесть ног, но глаз у обоих по два. Решение:

```
eyes = int(input())
legs = int(input())
if eyes >= 8:
    if legs == 8:
        print("паук")
    else:
        print("гребешок")
else:
    if legs == 6:
        print("жук")
    else:
        print("кошка")
```

Если вложенных условных операторов несколько, то, к какому из них относится else, можно понять по отступу. Отступ у else должен быть такой же, как у if, к которому он относится.

Конструкция «иначе-если»

В некоторых ситуациях необходимо осуществить выбор больше чем из двух вариантов, которые могут быть обработаны с помощью if-else.

Пример. Необходимо вывести словом название числа 1 или 2 или сообщить, что это другое число:

```
n = int(input())
if n == 1:
    print('Один')
elif n == 2:
    print('Два')
else:
    print('Другое число')
```

Здесь используется специальная конструкция elif, обозначающая "иначе, если", после которой записывается условие. Такая конструкция введена в язык Питон, потому что запись if-else приведет к увеличению отступа и ухудшению читаемости.

Пример. Дано натуральное число. Требуется определить, является ли год с данным номером високосным. Если год является високосным, то выведите YES, иначе выведите NO. Напомним, что в соответствии с григорианским календарем, год является високосным, если его номер кратен 4, но не кратен 100, или же если он кратен 400.

Формат ввода

Вводится одно натуральное число.

Формат вывода

Выведите ответ на задачу.

Тест 1
Входные данные:
4
Вывод программы:
YES

Решение:

```
a = int(input())
if (a % 4 == 0) and (a % 100 > 0) or (a % 400 == 0):
    print('YES')
else:
    print('NO')
```

Задание 1.

Даны два целых числа. Программа должна вывести число "1", если первое число больше второго, число "2", если второе больше первого или число "0", если они равны.

Формат ввода

Вводятся два целых числа.

Формат вывода

Выведите ответ на задачу.

Примечания

Эту задачу желательно решить с использованием каскадных инструкций if... elif... else.

Тест 1

Входные данные:

1
2

Вывод программы:

2

Задание 2.

Даны три целых числа. Найдите наибольшее из них (программа должна вывести ровно одно целое число).

Какое наименьшее число операторов сравнения ($>$, $<$, $>=$, $<=$) необходимо для решения этой задачи?

Воспользоваться функцией нахождения максимального из двух чисел $\max(a, b)$.

Формат ввода

Вводится три целых числа.

Формат вывода

Выведите ответ на задачу.

Тест 1

Входные данные:

1
2
3

Вывод программы:

3

Задание 3.

Дано три числа. Упорядочите их в порядке неубывания. Программа должна считывать три числа a, b, c , затем программа должна менять их значения так, чтобы стали выполнены условия $a \leq b \leq c$, затем программа выводит тройку a, b, c .

Формат ввода

Вводятся три числа.

Формат вывода

Выведите ответ на задачу.

Примечания

Дополнительные ограничения: нельзя использовать дополнительные переменные, то есть единственной допустимой операцией присваивания является обмен значений двух переменных типа $(a, b) = (b, a)$.

Можно делать присваивания любого количества переменных. Например, ввод в этой задаче можно оформить так:

```
a, b, c = int(input()), int(input()), int(input())
```

Тест 1

Входные данные:

1

2

1

Вывод программы:

1 1 2

Задание 4.

За многие годы заточения узник замка Иф проделал в стене прямоугольное отверстие размером $D \times E$. Замок Иф сложен из кирпичей, размером $A \times B \times C$. Определите, сможет ли узник выбрасывать кирпичи в море через это отверстие (очевидно, стороны кирпича должны быть параллельны сторонам отверстия).

Формат ввода

Программа получает на вход числа A, B, C, D, E .

Формат вывода

Программа должна вывести слово YES или NO.

Тест 1

Входные данные:

1

1

1

1

1

Вывод программы:

YES

Тест 2

Входные данные:

2

2

2

1

1

Вывод программы:

NO