**Цикл for**

**Цикл с параметром**

Все программы, которые мы писали ранее, представляли собой последовательность инструкций, которую нельзя было повторить несколько раз для разных значений. Однако используя цикл **for**, этого можно добиться.

Цикл **for**, также называемый циклом с параметром, в языке Python богат возможностями. В цикле **for** указывается переменная и множество значений, по которому будет пробегать переменная. Множество значений может быть задано списком.

**Пример:**

**for** color **in** 'red', 'green', 'blue':

 **print**(color)

В этом примере в качестве множества значений используется список цветов. Переменная color последовательно принимает значения 'red', 'green', 'blue'.

**Пример:**

Выведем на экран квадраты первых пяти простых чисел.

**for** i **in** 2, 3, 5, 7, 11:

 **print**(i \*\* 2)

**Функция range**

**Функция range()**

Как правило, при использовании цикла **for** необходимо, чтобы переменная принимала значения от некоторого начального значения до некоторого конечного. Чтобы реализовать это, удобно использовать функцию range().

**Пример:**

Чтобы вывести на экран числа от 00 до n−1n−1, можно использовать цикл **for** вместе с функцией range():

**for** i **in** range(n):

 **print**(i)

В качестве n может использоваться числовая константа или переменная. Если значение n равно нулю или отрицательное, то тело цикла не выполнится ни разу.

**Функция range() с двумя параметрами**

Если требуется задать цикл от некоторого числа aa до некоторого числа bb, то можно использовать функцию range() с двумя параметрами:

**for** i **in** range(a, b):

 **print**(i)

Индексная переменная i будет принимать значения от aa до b−1b−1, то есть первый параметр функции range(), вызываемой с двумя параметрами, задаёт начальное значение индексной переменной, а второй параметр — значение, которое индексная переменная принимать не будет. Если же b⩽ab⩽a, то цикл не будет выполнен ни разу.

**Пример:**

Чтобы просуммировать значения чисел от aa до bb можно воспользоваться следующей программой:

s = 0

**for** i **in** range(a, b):

 s += i

В этом примере переменная i принимает значения от aa до b−1b−1, и значение переменной s последовательно увеличивается на указанные значения.

**Функция range() с тремя параметрами**

Наконец, чтобы организовать цикл, в котором индексная переменная будет уменьшаться, необходимо использовать функцию range с тремя параметрами. Первый параметр задает начальное значение индексной переменной, второй параметр — значение, до которого будет изменяться индексная переменная (не включая его!), а третий параметр — величину изменения индексной переменной.

Более формально, цикл **for** i **in** range(a, b, d) при d > 0 задает значения индексной переменной i = a, i = a + d, i = a + 2 \* d и так для всех значений, для которых i < b.

Если же d < 0, то переменная цикла принимает значения i = a, i = a - |d|, i = a - 2 \* |d| и так для всех значений, для которых i > b.

**Пример:**

Cделать цикл по всем числам от 100100 до 11 можно при помощи следующей команды

range(100, 0, -1)

**Примеры применения цикла for**

Рассмотрим варианты применения цикла **for** на примерах задач.

**Задача. Числа, не делящиеся на три**

Выведите все числа от 11 до 9999, которые не делятся на 33.

**Решение**

**for** i **in** range(1, 100):

 **if** i % 3 != 0:

 **print**(i)

**Задача. Сумма**nn**чисел**

Необходимо вывести на экран сумму nn чисел. В первой строке входных данных содержится число nn — количество чисел. В следующих nn строках содержатся сами числа.

**Решение**

n = int(input())

s = 0

**for** i **in** range(n):

 a = int(input())

 s = s + a

**print**(s)

**Задача. Минимум из**nn**чисел**

Необходимо вывести на экран минимальное из nn чисел. В первой строке входных данных содержится число nn — количество чисел. В следующих nn строках содержатся сами числа.

**Решение**

n = int(input())

m = int(input())

**for** i **in** range(n - 1):

 a = int(input())

 **if** a < m:

 m = a

**print**(m)

Это решение можно немного упростить с помощью функции min:

n = int(input())

m = int(input())

**for** i **in** range(n - 1):

 m = min(m, int(input()))

**print**(m)

Пусть известно, что все числа, которые подаются на вход, не превосходят какой-либо константы. Например, 109109. Тогда можно не считывать отдельно первое из чисел:

n = int(input())

m = 10 \*\* 9

**for** i **in** range(n):

 m = min(m, int(input()))

**print**(m)