**Циклы**

В цикле наглядно видна сила автоматизации, когда мы запускаем процесс и он начинает делать многократные повторения.

Цикл напоминает штамповку. Мы один раз изготавливаем штамп и этим штампом изготавливаем много деталей. Этот процесс, правда, нужно как-то закончить. Нужен контроль за цикличностью. Цикличность, которая никогда не заканчивается, плоха, т.к. мы никогда не получим итоговый результат.

Цикл должен иметь конечное число итераций.

**Итерация** – однократное повторение действий, находящихся в цикле.

**Тело цикла** – действия, которые повторяются в цикле.

**Цикл с параметром**

Напечатаем все квадраты чисел из некоторого списка.

Пусть у нас есть список

L=[1,2,3,4,5]

**for** x in L:

print(x\*\*2)

Но если список должен быть длинным, то так неудобно.

Функция **range** генерирует некоторую арифметическую прогрессию.

**rande(N)** – список значений 0, 1,2, 3, …, N-1

**range(a,b)** – список значений от a до b-1 с шагом 1 (открытая правая граница)

**range(a,b,d)** – список значений от a до b-1 с шагом d (до a+n\*d<b). d м.б.-1

**Задача 1**. Напечатать все числа

а) от 1 до 100

**for** x **in range**(1,101)

print(x)

б) от 100 до 1

**for** x **in range**(100,0,-1)

print(x)

в) четные числа от 1 до 100

**for** x **in range**(2,101,2)

print(x)

**Обработка последовательностей**

Во всех задачах ниже программа получает на вход число N – количество чисел в последовательности (1 ≤ N ≤ 50), затем вводятся *N* чисел последовательности, каждое с новой строки..

**Ввод последовательности чисел** будет выглядеть так

n = **int**(**input**())

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

x = **int**(**input**())

**Задача 2.** Определить количество чисел, удовлетворяющих определенному условию.

Для подсчета количества вводится дополнительная переменная count, которой в начале присваивается 0. После ввода каждого числа нужно проверить, удовлетворяет оно условию или нет. Если удовлетворяет, то значение переменной count нужно увеличить на 1.

count = 0

n = **int**(**input**())

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

x = **int**(**input**())

**if** x удовлетворяет условию:

count += 1

**print**(count)

**Задача 3.** Определить присутствовало ли в последовательности число, удовлетворяющих определенному условию.

Данную задачу можно свести к предыдущей и посчитать количество чисел, удовлетворяющих условию. После цикла нужно только проверить переменную счетчик. Если она будет больше 0, то искомое число присутствовало, если она осталась равна 0, то ни одно значение последовательности не удовлетворяло условию.

Но можно не считать количество, а воспользоваться флагом (логической переменной). Вначале сделать логическую переменной равной False. После ввода каждого числа нужно проверить, удовлетворяет оно условию или нет. Если удовлетворяет, то значение логической переменной нужно сделать True.

**При этом ни в коем случае не нужно возвращать значение переменной в False, если попалось число не удовлетворяющее условию.**

is\_ok = **False**

n = **int**(**input**())

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

x = **int**(**input**())

**if** x удовлетворяет условию:

is\_ok = **True**

**if** is\_ok:

**print**('YES')

**else:**

**print**('NO')

**Задача 4.** Найти сумму чисел последовательности.

Для подсчета суммы вводится дополнительная переменная summ, которой в начале присваивается 0. После ввода каждого числа нужно добавить его к значению переменной summ.

summ = 0

n = **int**(**input**())

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

x = **int**(**input**())

summ += x

**print**(summ)

**Задача 5.** Найти произведение чисел последовательности.

Задача похожа на предыдущую. Только начальное значение переменной prod должно быть 1. После ввода каждого числа нужно помножать на него к значению переменной prod.

prod = 1

n = **int**(**input**())

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

x = **int**(**input**())

prod \*= x

**print**(prod)

**Задача 6.** Найти сумму (произведение) чисел последовательности, удовлетворяющих условию.

Отличие от предыдущих задач только в том, что перед суммированием (домножением) нужно проверить, удовлетворяет ли число условию. В конце по значению переменной sum (prod) можно узнать были ли числа, удовлетворяющие условию.

summ = 0

n = **int**(**input**())

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

x = **int**(**input**())

**if** x удовлетворяет условию:

summ += x

**if** sum == 0:

**print**('NO')

**else:**

**print**(summ)

**Задача 7.** Найти среднее арифметическое чисел последовательности, удовлетворяющих условию.

Нужно найти сумму и количество чисел, удовлетворяющих условию и поделить сумму на количество.

summ = 0

count = 0

n = **int**(**input**())

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

x = **int**(**input**())

**if** x удовлетворяет условию:

summ += x

count += 1

**if** count == 0:

**print**('NO')

**else:**

**print**(summ / count)

**Задача 8.** Найти минимальное число в последовательности. Числа в последовательности из диапазона от -1000 до 1000.

Нужно ввести переменную minn, в которой будем искать минимальное число. В начале туда нужно положить очень большое число 1001. Каждое введенное число нужно сравнить с этой переменной. Если введенное число меньше значения в minn, то нужно заменить это значение на введенное число. **Если введенное число не меньше minn, то ничего с переменной minn делать не нужно.**

minn = 1001

n = **int**(**input**())

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

x = **int**(**input**())

**if** x< minn:

minn = x

**print**(minn)

Максимальное значение ищется аналогично. Нужно только в начале положить в переменную очень маленькое число -1001 и знак < заменить на >.

**Задача 9.** Найти минимальное число в последовательности и его номер. Числа в последовательности из диапазона от -1000 до 1000.

В данной задаче помимо значения минимального элемента нужно хранить его номер i\_min.

minn = 1001

i\_min = -1

n = **int**(**input**())

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

x = **int**(**input**())

**if** x< minn:

minn = x

i\_min = i

**print**(minn, i\_min)

Если элементы нумеруются с 1, то номер будет на 1 больше, т.е. i\_min + 1.

При этом, если в последовательности присутствует несколько чисел с минимальным значением, то будет найден номер самого первого числа.

Чтобы найти номер самого последнего числа в последовательности, значение которого равно минимальному, нужно в условие добавить нестрогое неравенство

**if** x<= minn:

**Задача 9.** Найти минимальное число в последовательности и его номер, среди чисел, удовлетворяющих условию. Числа в последовательности из диапазона от -1000 до 1000.

В данной задаче добавляется только проверка условия.

minn = 1001

i\_min = -1

n = **int**(**input**())

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

x = **int**(**input**())

**if** x удовлетворяет условию **and** x< minn:

minn = x

i\_min = i

**if** minn == 100**1:**

**print**('NO')

**else**:

**print**(minn, i\_min)

Если элементы нумеруются с 1, то номер будет на 1 больше, т.е. i\_min + 1.

**Задача 10.** Найти минимальное число в последовательности и его номер, среди чисел, удовлетворяющих условию. Числа в последовательности могут быть произвольными.

В данной задаче в minn помещается значение, которое не удовлетворяет условию (или в i\_min кладется -1). А далее значения изменяются, если число удовлетворяет условию и, либо это первое значение, которое удовлетворяет условию (в minn значение, которое не удовлетворяет условию, или i\_min = -1), или это значение меньше предыдущего найденного минимума.

Рассмотрим пример программы, где в i\_min положим -1.

minn = 1 # любое число

i\_min = -1

n = **int**(**input**())

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

x = **int**(**input**())

**if** x удовлетворяет условию **and** (ш\_ьшт == -1 **щк** x< minn):

minn = x

i\_min = i

**if** i\_min == -1**:**

**print**('NO')

**else**:

**print**(minn, i\_min)

Если элементы нумеруются с 1, то номер будет на 1 больше, т.е. i\_min + 1.