12. Опишите синтаксис создания операторов break и continue в языке Python.

Инструкция break передает управление за пределы самого внутреннего цикла, в котором она встречается, то есть прерывает выполнение цикла.

Цикл while совместно с оператором break удобно использовать для получения неопределенного заранее количества данных от пользователя

Точно так же, как и в случае с циклом while, инструкция for поддерживает инструкции break и continue, а также необязательное предложение else.

Cинтаксис его использования будет выглядеть примерно так:

i = 1

while True:

if i > 100:

break

print(i)

i += 1

Инструкция continue передает управление в начало цикла. Как правило, инструкции break и continue используются внутри инструкций if, чтобы в зависимости от складывающихся условий изменять поведение цикла.

Cинтаксис его использования будет выглядеть примерно так:

for i in range(l, 101):

if 4 < i < 11:

continue

print(i)

52. Опишите методы генерирования списков в языке Python.

Самым простым способом создания списка является обычное присваивание ему необходимых значений или объектов. В том случае, когда элементов последовательности немного, их можно просто перечислить один за другим. Но если их количество переваливает за десяток, можно использовать генератор списка. **Данная конструкция обеспечивает его автоматическое заполнение, исходя из определенных инструкций.**

Один из часто используемых способов генерации списков в Python — с условием. Данные конструкции языка Python **позволяют задавать определенные условия для выполнения такой операции**.

Предположим, например, что нам требуется получить список високосных годов в определенном диапазоне. Для начала мы могли бы использовать такой цикл:

leaps =[]

for year in range(1900, 1940):

if (year % 4 == 0 and year % 100 != 0) or (year % 400 ==0):

leaps.append(year)

Но можно использовать и такой способ: заключить выражение и цикл с дополнительным условием в квадратные скобки, в котором цикл используется для создания элементов списка, а условие используется для исключения нежелательных элементов. В результате мы получаем генератор списка в простейшем его виде который записывается, как показано ниже:

[item for item in iterable]

Это выражение вернет список всех элементов объекта iterable и семантически ничем не отличается от выражения list(iterable).

Интересными генераторы списков делают две особенности - они могут использоваться как выражения, и они допускают включение условной инструкции, вследствие чего мы получаем две типичные синтаксические конструкции использования генераторов списков:

[expression for item in iterable]

[expression for item in iterable if condition]

Вторая форма записи эквивалентна циклу:

temp =[]

for item in iterable:

if condition:

temp.append(expressiofl)

Обычно выражение expression является либо самим элементом item, либо некоторым выражением с его участием. Конечно, генератору списков не требуется временная переменная temp[ ], которая необходима в версии с циклом for ... in.

65. Напишите программу, которая определяет число отрицательных чисел во введенной с клавиатуры последовательности (длина последовательности неограниченна).

# coding: utf-8

print('Task 65 Cipher 29 Second year of Mogilev State Polytechnic College')

max = +1

max = int(input("Enter the maximum positive number of the sequence: ")) # ввод максимума

t = 0.0

i = 0

while True: #цикл ввода последовательности и проверка на отрицательность–увеличить счетчик при совпадении- если число больше максимума выход из цикла

t = float(input("Input the sequence : "));

if t <= 0:

i += 1

if t >= max:

break

print("The number of Negative numbers - ", i)



76. Решите задачу. Дан двумерный массив. Заполните его по строкам с клавиатуры и определите:

- количество строк, не содержащих ни одного нулевого элемента;

- максимальное из чисел, в заданной строке массива.

rows = int(input('Rows: '))

cols = int(input('Columns: '))

arr = [[int(input()) for i in range(cols)] for j in range(rows)]

for row in arr:

print(row)

print('Count rows without zeros: ')

print(len([i for i in arr if 0 not in i]))

n = int(input('Enter row index where you wish to locate the maximum: ')) # индекс массива

print(f'Maximum in row: {max(arr[n])}')

