Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМУПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра Инновационных

технологий (ФИТ)

Отчет

по лабораторной работе №7

**«ЦИКЛЫ FOR И WHILE»**

по дисциплине «Алгоритмы и структуры данных»

Студент гр. 023

\_\_\_\_\_\_\_\_\_\_\_\_ Иптышев В.А

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2025 г.

Доцент кафедры УИ

\_\_\_\_\_\_\_\_\_\_ Лобода Ю.О.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2025 г.

Томск 2025

**Содержание**

[Введение 3](#_Toc192806723)

[I ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИМЕНЕНИЯ УСЛОВНОГО ОПЕРАТОРА В PYTHON 4](#_Toc192806724)

[II ПРАКТИЧЕСКАЯ ЧАСТЬ 7](#_Toc192806725)

[2.1. Решения задач на тему «Выбор из двух» 7](#_Toc192806726)

[2.2 Решения задач на тему «Логические операции» 10](#_Toc192806727)

[2.3 Решения задач на тему «Вложенные и каскадные условия» 12](#_Toc192806728)

[2.4 Итоговая работа на условный оператор 16](#_Toc192806729)

[Заключение 19](#_Toc192806730)

# Введение

В жизни часто встречаются ситуации, где приходится выполнять однотипные, монотонные действия. Например, сортировка деталей, выставление оценок в журнале, рассада моркови на грядке – все эти занятия требуют повторения определенных операций. Для решения этой задачи были придуманы циклы. В любой ситуации, где приходится что-либо повторять, они являются хорошими помощниками. Использование циклов может сильно сократить написанный код, и сделать его более понятным и лаконичным для чтения.

В Python существует две основных разновидности циклов: *for* и *while*. Цикл *for* замечательно работает, если мы заранее знаем, сколько повторений (итераций) нам требуется сделать. В противном случае стоит использовать *while*.

**Целью** лабораторной работы является изучение циклов в Python.

**Задачи**:

1. Изучить теорию по циклам в Python;
2. Решить задачи на тему циклы;
3. Сделать вывод по применению циклов в Python.

# I ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИМЕНЕНИЯ ЦИКЛОВ В PYTHON

Чтобы написать цикл с заранее известным количеством повторений можно использовать оператор *for*.

Структура цикла *for* в Python выглядит так:

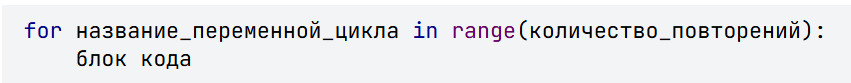


Рисунок 1.1 — структура цикла *for* в Python.

Рассмотрим следующий программный код:

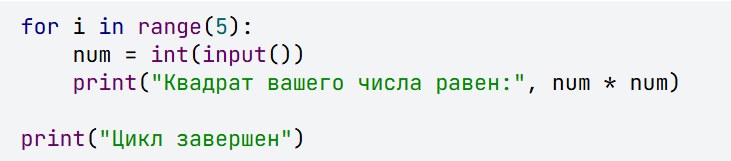


Рисунок 1.2 — код, выполняющий некую логику с использованием цикла *for*.

Такая программа считывает 5 чисел и выводит на экран их квадраты вместе с поясняющей надписью. Поскольку вторая и третья строки выделены отступом, Python считает, что это тело цикла, которое выполняется 5 раз. Четвертая строка не содержит отступа, поэтому не является частью цикла и будет выполнена всего один раз, после того как цикл завершится.

Цикл *for* замечательно работает, если мы заранее знаем, сколько повторений (итераций) нам потребуется сделать. Но иногда нужно, чтобы цикл выполнялся до наступления некоторого события, и количество итераций в этом случае заранее оценить просто невозможно. И здесь на помощь приходит цикл *while*.

Структура цикла *while* в Python выглядит так:

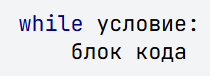


Рисунок 1.2 — структура цикла while.

Двоеточие (:) в конце строки с инструкцией while сообщает Python, что дальше находится блок команд. В блок входят все строки, расположенные с отступом от строки с инструкцией while, вплоть до следующей строки без отступа.

Блок команд, который выполняется в цикле while, называется телом цикла.

Рассмотрим код, использующий цикл while, который распечатает 10 раз слово Привет:

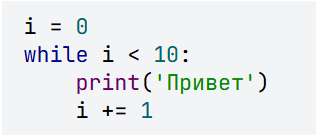


Рисунок 1.2 — код, печатающий 10 раз слово «Привет».

Такой код можно легко заменить циклом for, поскольку мы заранее знаем, сколько раз нужно выполнить тело цикла. Однако так бывает не всегда.

Напишем программу, которая считывает числа и выводит их квадраты, пока не будет введено −1. При такой постановке задачи мы не можем воспользоваться циклом for, так как не знаем, сколько чисел будет предшествовать числу −1.

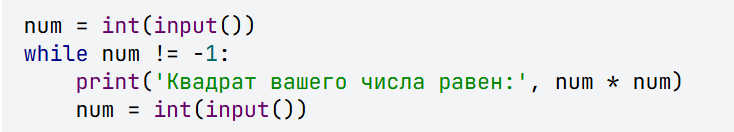


Рисунок 1.2 — код, печатающий квадраты введенного числа до того момента, пока не было введено -1.

# II ПРАКТИЧЕСКАЯ ЧАСТЬ

## 2.1. Решения задач на тему «Цикл for»

«Python is awesome!».

Напишите программу, которая выводит текст «Python is awesome!» (без кавычек) 10 раз

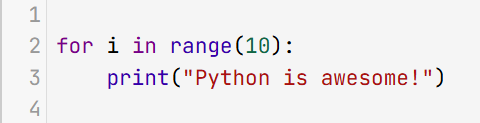


Рисунок 2.1 — решение задачи «Python is awesome!».

«Последовательность символов».

Напишите программу, которая использует ровно три цикла for для печати следующей последовательности символов:

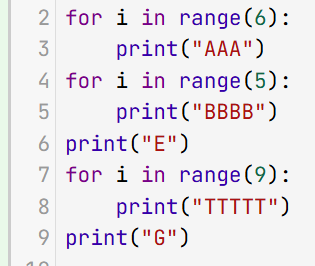


Рисунок 2.2 — решение задачи «Последовательность символов».

«Повторяй за мной 1».

Дано предложение и количество раз, сколько его надо повторить. Напишите программу, которая повторяет данное предложение нужное количество раз.

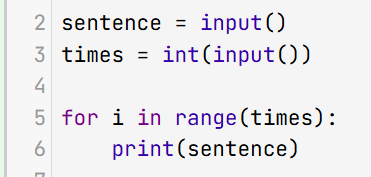


Рисунок 2.3 — решение задачи «Повторяй за мной 1».

«Звездный прямоугольник».

На вход программе подаётся натуральное число n (n∈ [1;20]).

Напишите программу, которая печатает звёздный прямоугольник размерами n×19.

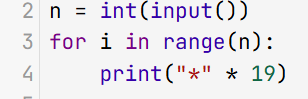


Рисунок 2.4 — решение задачи «Звездный прямоугольник».

«Повторяй за мной 2».

Напишите программу, которая считывает одну строку текста и выводит

10 строк, пронумерованных от 9, каждая, с указанной строкой текста.

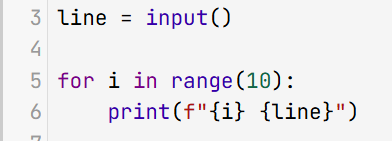


Рисунок 2.5 — решение задачи «Повторяй за мной 2».

«Звездный треугольник».

На вход программе подаётся натуральное число n(n≥2) – катет прямоугольного равнобедренного треугольника.

Напишите программу, которая выводит звёздный треугольник в соответствии с примером.

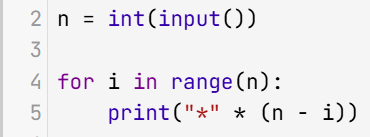


Рисунок 2.6 — решение задачи «Звездный треугольник».

«Популяция».

На вход программе подаются три натуральных числа m,p,n*m*,*p*,*n*:

* m:*m*: стартовое количество организмов;
* p:*p*: среднесуточное увеличение в %%;
* n:*n*: количество дней для размножения.

Напишите программу, которая предсказывает размер популяции организмов с 11-го по n*n*-й день (включительно). Программа должна выводить номер дня, а затем через пробел размер популяции в этот день.

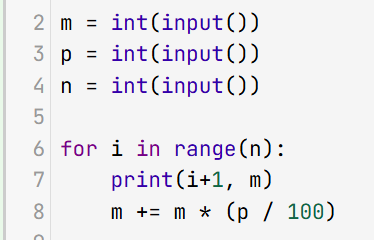


Рисунок 2.7 — решение задачи «Популяция».

«Наибольшее и наименьшее».

Напишите программу, которая находит наименьшее и наибольшее из пяти чисел и выводит текст в следующем формате:

Наименьшее число = <наименьшее число>

Наибольшее число = <наибольшее число>

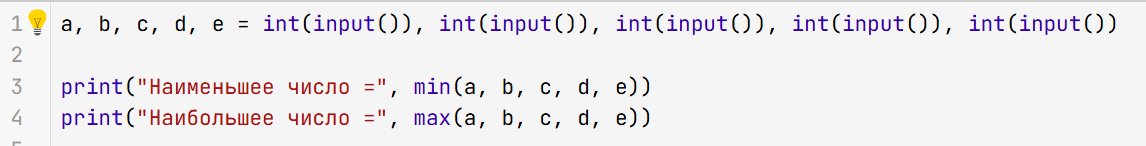


Рисунок 2.8 — решение задачи «Наибольшее и наименьшее».

«Сортировка трёх».

Напишите программу, которая упорядочивает три числа от большего к меньшему.

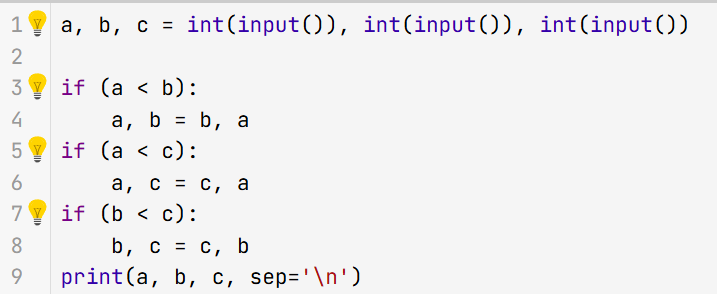


Рисунок 2.9 — решение задачи «Сортировка трёх».

## 2.2 Решения задач на тему «Цикл for: функция range».

«Последовательность чисел 1».

Даны два целых числа m*m* и n*n* (m≤n*m*≤*n*). Напишите программу, которая выводит все целые числа от m*m* до n*n* включительно.

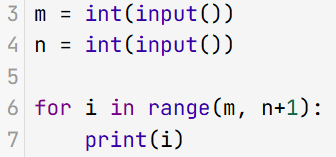


Рисунок 2.10 — решение задачи «Последовательность чисел 1».

«Последовательность чисел 2».

Даны два целых числа m*m* и n*n*. Напишите программу, которая выводит все целые числа от m*m* до n*n* включительно в порядке возрастания, если m<n*m*<*n*, или в порядке убывания в противном случае.

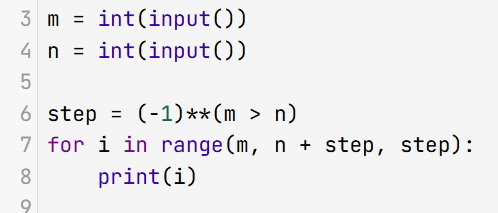


Рисунок 2.11 — решение задачи «Последовательность чисел 2».

«Последовательность чисел 3».

Даны два целых числа m*m* и n*n* (m>n)(*m*>*n*). Напишите программу, которая выводит все нечётные целые числа от m*m* до n*n* (включительно) в порядке убывания.

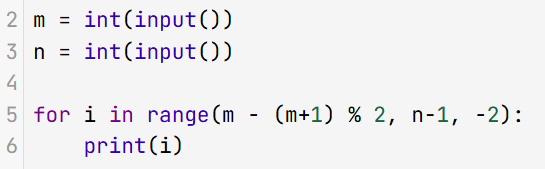


Рисунок 2.12 — решение задачи «Последовательность чисел 3».

«Последовательность чисел 4».

Даны два натуральных числа m*m* и n*n* (m≤n*m*≤*n*). Напишите программу, которая выводит все целые числа от m*m* до n*n* (включительно), удовлетворяющие **хотя бы одному** из условий:

* число кратно 1717
* число оканчивается на 99
* число кратно 33 и 55 одновременно

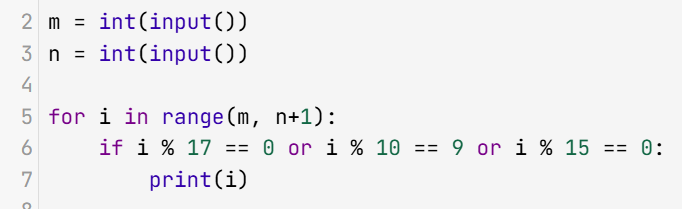


Рисунок 2.12 — решение задачи «Последовательность чисел 4».

«Таблица умножения».

Дано натуральное число n*n*. Напишите программу, которая выводит таблицу умножения на n*n* (от 11 до 1010 включительно).

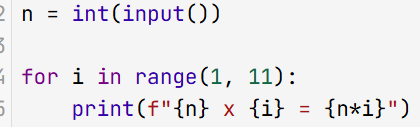


Рисунок 2.12 — решение задачи «Таблица умножения».

## 2.3 Решения задач на тему «Частые сценарии».

«Количество чисел».

На вход программе подаются два целых числа a*a* и b*b* (a≤b)(*a*≤*b*). Напишите программу, которая подсчитывает количество чисел в диапазоне от a*a* до b*b* (включительно), куб которых оканчивается на 44 или 99.

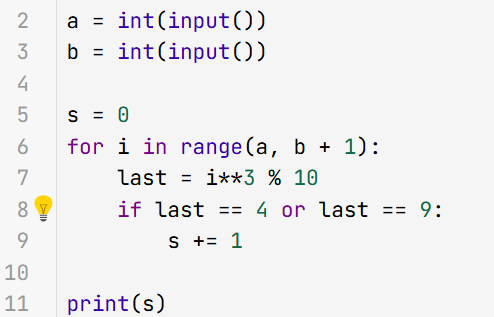


Рисунок 2.13 — решение задачи «Количество чисел».

«Асимптотическое приближение».

На вход программе подаётся натуральное число n*n*. Напишите программу, которая вычисляет значение выражения

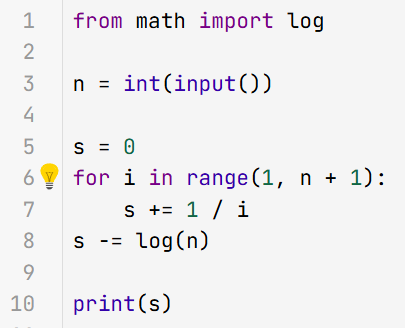


Рисунок 2.14 — решение задачи «Асимптотическое приближение».

«Сумма чисел 2».

На вход программе подаётся натуральное число n*n*. Напишите программу, которая подсчитывает сумму тех чисел от 11 до n*n* (включительно), квадрат которых оканчивается на 2,52,5 или 88.

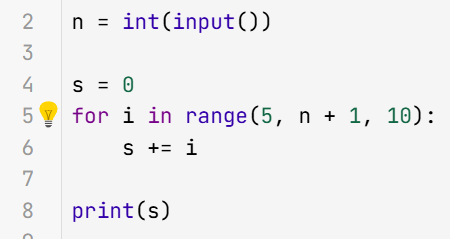


Рисунок 2.15 — решение задачи «Сумма чисел 2».

«Факториал!».

На вход программе подаётся натуральное число n*n*. Напишите программу, которая вычисляет n!*n*!.

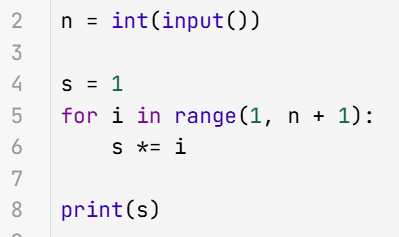


Рисунок 2.16 — решение задачи «Факториал!».

«Сумма делителей».

На вход программе подаётся натуральное число n*n*. Напишите программу, которая вычисляет сумму всех его делителей.

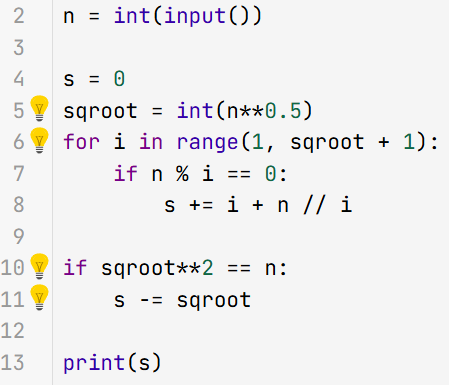


Рисунок 2.17 — решение задачи «Сумма делителей».

«Наибольшие числа».

На вход программе подаются натуральное число n(n≥2)*n*(*n*≥2), а затем n*n* различных натуральных чисел последовательности, каждое на отдельной строке. Напишите программу, которая выводит наибольшее и второе наибольшее число последовательности.

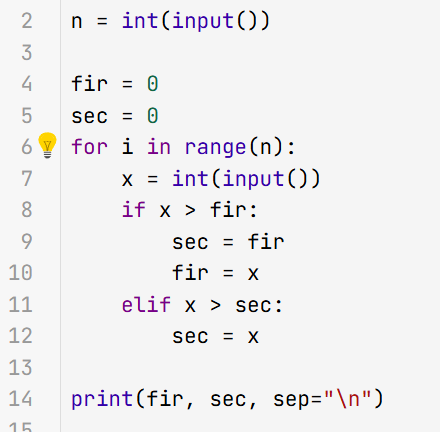


Рисунок 2.18 — решение задачи «Наибольшие числа».

«Only even numbers».

Напишите программу, которая считывает последовательность из 1010 целых чисел и определяет, является ли каждое из них чётным или нет.

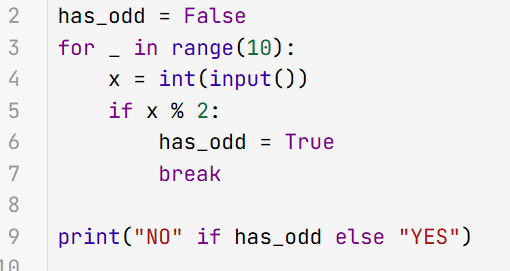


Рисунок 2.19 — решение задачи «Only even numbers».

«Последовательность Фибоначчи».

Напишите программу, которая считывает натуральное число n*n* и выводит первые n*n* чисел последовательности Фибоначчи.

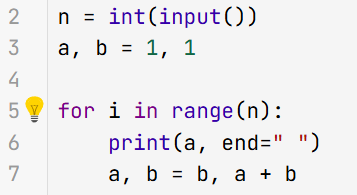


Рисунок 2.20 — решение задачи «Последовательность Фибоначчи».

## 2.4 Решения задач на тему «Цикл while».

«Количество членов».

На вход программе подаётся последовательность слов, каждое слово на отдельной строке. Концом последовательности является одно из трёх слов: «**стоп»**, «**хватит»**, «**достаточно»** (без кавычек). Сами эти слова в последовательность не входят, лишь символизируя её окончание. Напишите программу, которая выводит общее количество членов данной последовательности.

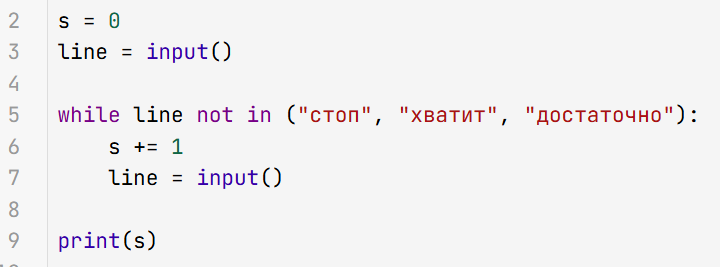


Рисунок 2.21 — решение задачи «Количество членов».

«Пока делимся».

На вход программе подаётся последовательность целых чисел делящихся на 77, каждое число на отдельной строке. Концом последовательности является любое число, **не делящееся** на 77 (само это число в последовательность не входит, лишь символизируя её конец). Напишите программу, которая выводит члены данной последовательности.

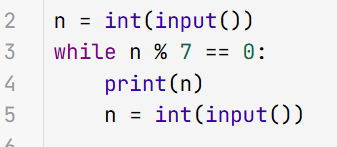


Рисунок 2.22 — решение задачи «Пока делимся».

«Сумма чисел».

На вход программе подаётся последовательность целых чисел, каждое число на отдельной строке. Признаком окончания последовательности является любое отрицательное число, при этом в саму последовательность оно не входит. Напишите программу, которая выводит сумму всех членов данной последовательности.

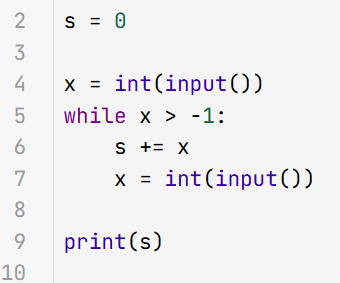


Рисунок 2.23 — решение задачи «Сумма чисел».

«Количество пятерок».

На вход программе подаётся последовательность целых чисел от 11 до 55, характеризующее оценку ученика, каждое число на отдельной строке. Концом последовательности является любое неположительное число либо число, большее 55. Напишите программу, которая выводит количество пятерок.

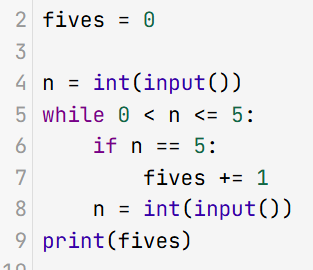


Рисунок 2.24 — решение задачи «Количество пятерок»

«Ведьмаку заплатите чеканной монетой».

Всем известно, что ведьмак способен одолеть любых чудовищ, однако его услуги обойдутся недешево. К тому же ведьмак не принимает купюры, он принимает только чеканные монеты. В мире ведьмака существуют монеты с номиналами 1,5,10,251,5,10,25.

Напишите программу, которая определяет, какое минимальное количество чеканных монет нужно заплатить ведьмаку.

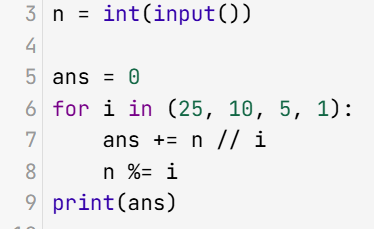


Рисунок 2.25 — решение задачи «Ведьмаку заплатите чеканной монетой»

## 2.5 Решения задач на тему «Цикл while: обработка цифр числа».

«Обратный порядок 1».

Дано натуральное число. Напишите программу, которая выводит его цифры в столбик в обратном порядке.

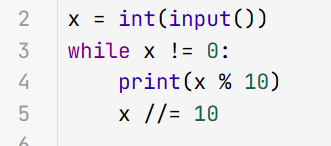


Рисунок 2.26 — решение задачи «Обратный порядок 1».

«max и min».

Дано натуральное число n(n≥10)*n*(*n*≥10). Напишите программу, которая определяет его максимальную и минимальную цифры и выводит текст в следующем формате:

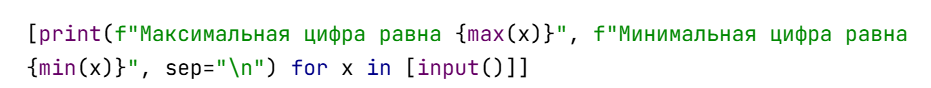


Рисунок 2.27 — решение задачи «max и min».

«Все вместе»

Дано натуральное число. Напишите программу, которая вычисляет:

* сумму его цифр;
* количество цифр в нем;
* произведение его цифр;
* среднее арифметическое его цифр;
* его первую цифру;
* сумму его первой и последней цифры.

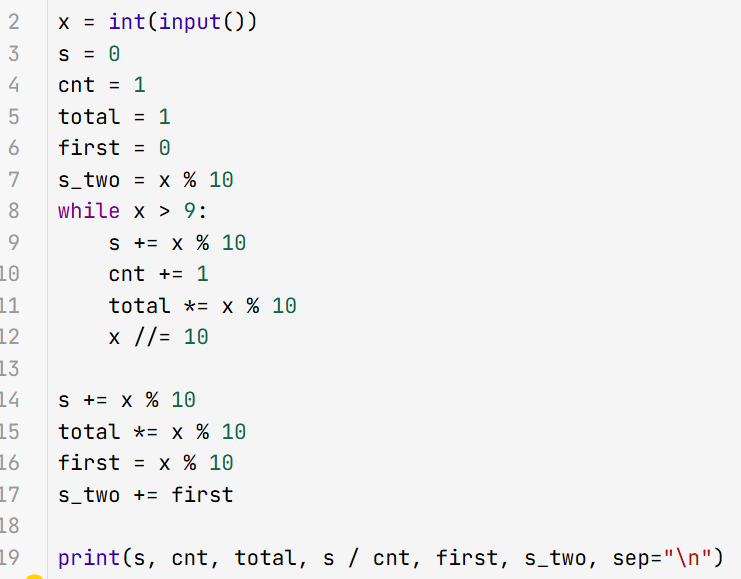


Рисунок 2.28 — решение задачи «Все вместе».

«Одинаковые цифры»

Дано натуральное число. Напишите программу, которая определяет, состоит ли указанное число из одинаковых цифр.



Рисунок 2.29 — решение задачи «Одинаковые цифры».

«Упорядоченные цифры»

Дано натуральное число. Напишите программу, которая определяет, является ли последовательность его цифр при просмотре справа налево упорядоченной по неубыванию.



Рисунок 2.30 — решение задачи «Упорядоченные цифры».

## 2.6 Решения задач на тему «Break, continue и else».

«Наименьший делитель».

На вход программе подаётся число n(n>1)*n*(*n*>1). Напишите программу, которая выводит его наименьший отличный от 11 делитель.

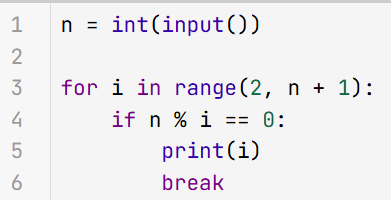


Рисунок 2.31 — решение задачи «Наименьший делитель».

«Следуй правилам».

На вход программе подаётся натуральное число n*n*. Напишите программу, которая выводит числа от 11 до n*n* включительно за исключением:

* чисел от 55 до 99 включительно;
* чисел от 1717 до 3737 включительно;
* чисел от 7878 до 8787 включительно.

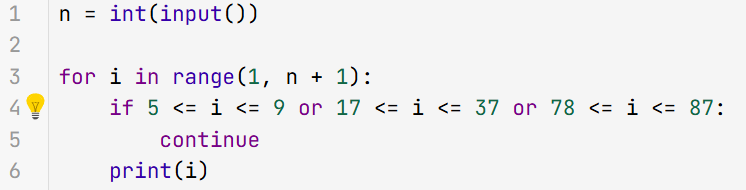


Рисунок 2.32 — решение задачи «Следуй правилам».

## 2.7 Решения задач на тему «Поиск ошибок и ревью кода».

«Ревью кода-1».

На обработку поступает последовательность из 1010 целых чисел (каждое на отдельной строке). Нужно написать программу, которая выводит на экран количество неотрицательных чисел последовательности и их произведение. Если неотрицательных чисел нет, требуется вывести на экран «NO» (без кавычек). Программист торопился и написал программу неправильно.

Найдите все ошибки в этой программе (их ровно 4). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк.

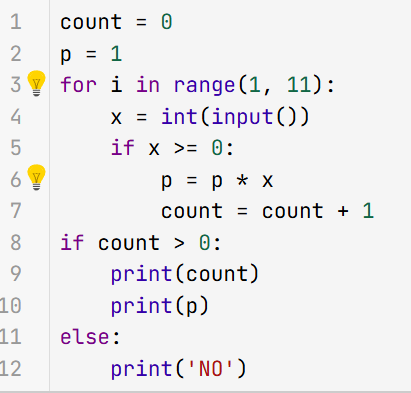


Рисунок 2.33 — решение задачи «Ревью кода-1».

«Ревью кода-2».

На обработку поступает последовательность из 1010 целых чисел (каждое на отдельной строке). Известно, что вводимые числа по абсолютной величине не превышают 106106. Нужно написать программу, которая выводит на экран сумму всех отрицательных чисел последовательности и максимальное отрицательное число в последовательности. Если отрицательных чисел нет, требуется вывести на экран «NO» (без кавычек). Программист торопился и написал программу неправильно.

Найдите все ошибки в этой программе (их ровно 5). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк.

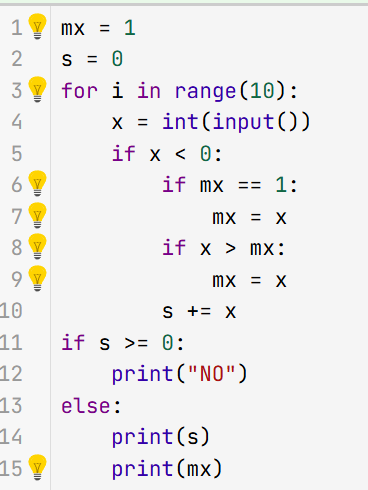


Рисунок 2.34 — решение задачи «Ревью кода-2».

«Ревью кода-3».

На обработку поступает последовательность из 77 целых чисел (каждое на отдельной строке). Нужно написать программу, которая подсчитывает и выводит сумму всех чётных чисел последовательности. Если таких чисел нет, программа должна вывести 00. Программист торопился и написал программу неправильно.

Найдите все ошибки в этой программе (их ровно 4). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк.

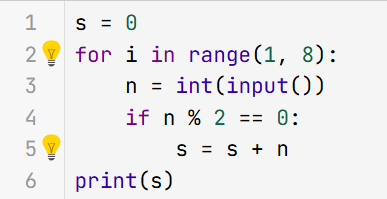


Рисунок 2.35 — решение задачи «Ревью кода-3».

«Ревью кода-4».

На обработку поступает натуральное число. Нужно написать программу, которая выводит на экран максимальную цифру числа, кратную 33. Если в числе нет цифр, кратных 33, требуется на экран вывести «NO» (без кавычек). Программист торопился и написал программу неправильно.

Найдите все ошибки в этой программе (их ровно 5). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк.

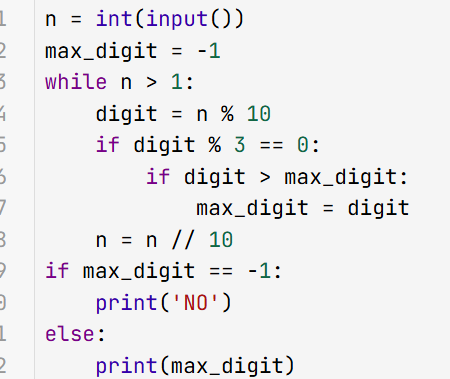


Рисунок 2.35 — решение задачи «Ревью кода-4».

## 2.8 Решения задач на тему «Вложенные циклы. Часть 1».

«Таблица-1».

Дано натуральное число n(n≤ 9)*n*(*n*≤ 9). Напишите программу, которая печатает таблицу размером n×5*n*×5, где в i*i*-ой строке указано число i*i* (числа отделены одним пробелом).



Рисунок 2.36 — решение задачи «Таблица-1».

«Таблица-2».

Дано натуральное число n(n≤9)*n*(*n*≤9). Напишите программу, которая печатает таблицу сложения для всех чисел от 11 до n*n* (включительно) в соответствии с примером.

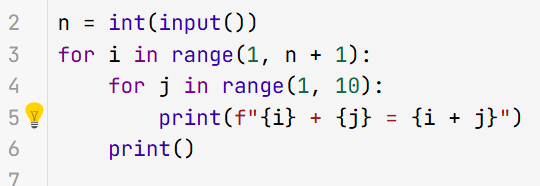


Рисунок 2.37 — решение задачи «Таблица-2».

«Звездный треугольник».

Дано нечётное натуральное число n*n*. Напишите программу, которая печатает [равнобедренный](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B2%D0%BD%D0%BE%D0%B1%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D1%82%D1%80%D0%B5%D1%83%D0%B3%D0%BE%D0%BB%D1%8C%D0%BD%D0%B8%D0%BA) звёздный треугольник с основанием, равным n*n*, в соответствии с примером:

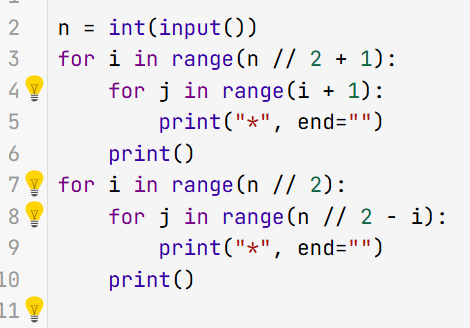


Рисунок 2.38 — решение задачи «Звездный треугольник».

«Численный треугольник 1».

Дано натуральное число n*n*. Напишите программу, которая печатает численный треугольник в соответствии с примером:

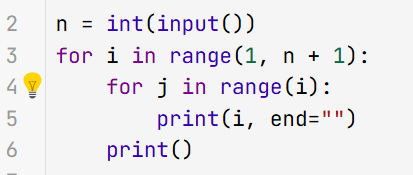


Рисунок 2.39 — решение задачи «Численный треугольник 1».

## 2.9 Решения задач на тему «Вложенные циклы. Часть 2».

«Численный треугольник 2».

Дано натуральное число n*n*. Напишите программу, которая печатает численный треугольник с высотой, равной n*n*, в соответствии с примером:

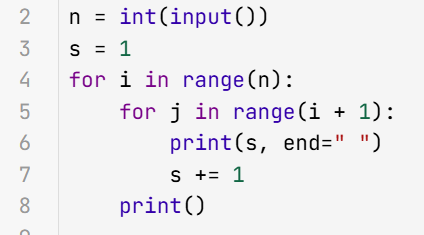


Рисунок 2.40 — решение задачи «Численный треугольник 2».

«Численный треугольник 3».

Дано натуральное число n*n*. Напишите программу, которая печатает численный треугольник с высотой, равной n*n*, в соответствии с примером:

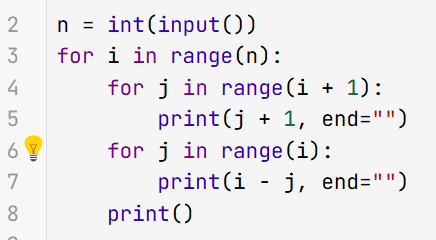


Рисунок 2.41 — решение задачи «Численный треугольник 3».

«Делители 1».

На вход программе подаются два натуральных числа a*a* и b*b* (a< b*a*< *b*). Напишите программу, которая находит натуральное число из отрезка [a;b][*a*;*b*] (от a*a* до b*b* включительно) с максимальной **суммой** делителей. Если чисел с максимальной суммой делителей несколько, то искомым числом является наибольшее из них. Ваша программа должна выводить ответ в следующем формате:

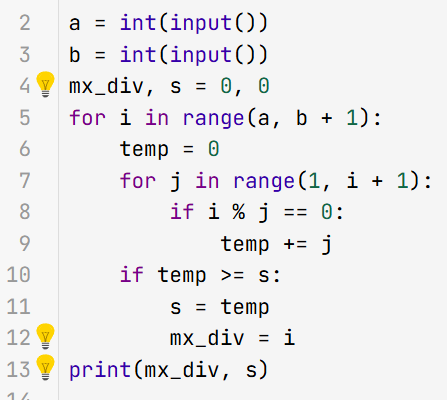


Рисунок 2.42 — решение задачи «Делители 1».

«Цифровой корень».

Цифровым корнем числа n*n*называется число, получающееся следующим образом: вычисляется сумма цифр числа n*n*, затем сумма цифр у получившегося числа и так далее, пока не получится однозначное число. Например, цифровой корень числа 98759875 равен 22:

На вход программе подаётся натуральное число n*n*. Напишите программу, которая находит цифровой корень данного числа.

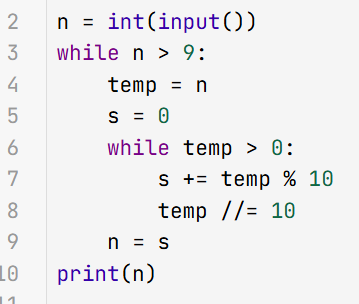


Рисунок 2.43 — решение задачи «Цифровой корень».

«Сумма факториалов».

Дано натуральное число n*n*. Напишите программу, которая выводит значение суммы:

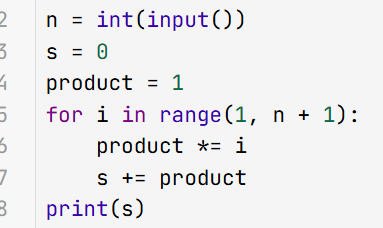


Рисунок 2.44 — решение задачи «Сумма факториалов».

«Простые числа».

На вход программе подается два натуральных числа a*a* и b(a<b)*b*(*a*<*b*). Напишите программу, которая находит все простые числа от a*a* до b*b* включительно.

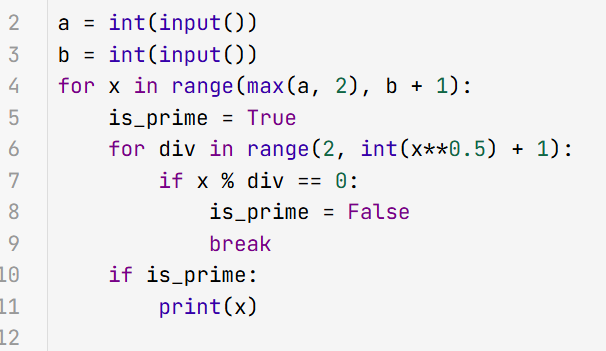


Рисунок 2.45 — решение задачи «Простые числа».

# Заключение

В рамках лабораторной работы была достигнута цель изучения циклов. Лабораторная работа позволила углубить понимание циклов в Python’е.

В заключение лабораторной работы были изучены тонкости использования циклов в практических задачах. Таким образом, работа стала значимым шагом на пути к пониманию языка программирования Python.

"Python is awesome 🐍"

Напишите программу, которая выводит текст «Python is awesome!» (без кавычек) $10$ раз.

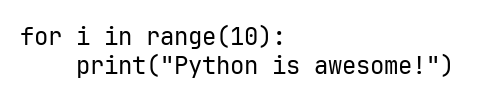


Рисунок 2.1 — решение задачи "Python is awesome 🐍".

"Последовательность символов"

Напишите программу, которая использует ровно три цикла for для печати следующей последовательности символов:

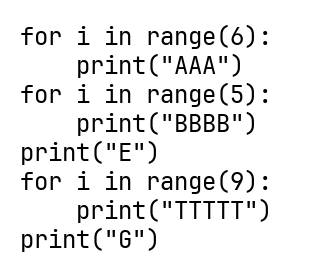


Рисунок 2.2 — решение задачи "Последовательность символов".

"Повторяй за мной 1"

Дано предложение и количество раз, сколько его надо повторить. Напишите программу, которая повторяет данное предложение нужное количество раз.

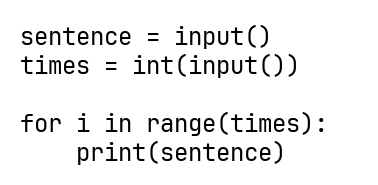


Рисунок 2.3 — решение задачи "Повторяй за мной 1".

"Звёздный прямоугольник ⭐"

На вход программе подаётся натуральное число \(n \, (n \in [1; \, 20])\).  
Напишите программу, которая печатает звёздный прямоугольник размерами \(n \times 19\).

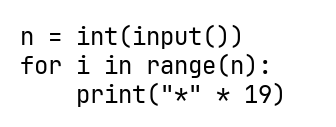


Рисунок 2.4 — решение задачи "Звёздный прямоугольник ⭐".

"Повторяй за мной 2"

Напишите программу, которая считывает одну строку текста и выводит $10$ строк, пронумерованных от $0$ до $9$ каждая, с указанной строкой текста.

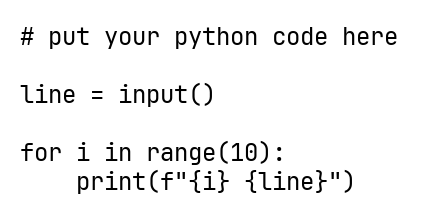


Рисунок 2.5 — решение задачи "Повторяй за мной 2".

"Квадрат числа"

На вход программе подаётся натуральное число $n$. Напишите программу, которая для каждого из чисел от $0$ до $n$ (включительно) выводит текст в следующем формате:

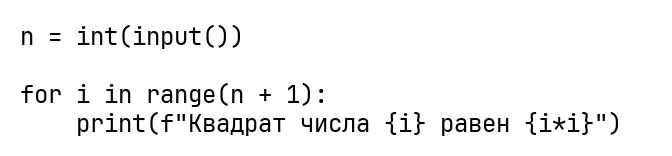


Рисунок 2.6 — решение задачи "Квадрат числа".

"Звёздный треугольник ⭐"

На вход программе подаётся натуральное число $n \, (n \ge 2)$ – катет прямоугольного равнобедренного треугольника.  
Напишите программу, которая выводит звёздный треугольник в соответствии с примером.

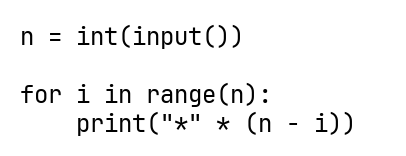


Рисунок 2.7 — решение задачи "Звёздный треугольник ⭐".

"Популяция 🦠"

На вход программе подаются три натуральных числа \(m, \, p, \, n\):  
Напишите программу, которая предсказывает размер популяции организмов с $1$-го по $n$-й день (включительно). Программа должна выводить номер дня, а затем через пробел размер популяции в этот день.

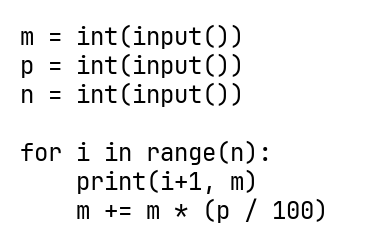


Рисунок 2.8 — решение задачи "Популяция 🦠".