

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет «СТАНКИН»**

**(ФГБОУ ВО МГТУ «СТАНКИН»)**

|  |  |
| --- | --- |
| **Институт**  информационных технологий | **Кафедра**  информационных систем |

**Отчет по лабораторной работе №2**

по дисциплине «**Веб-программирование**»

на тему: “Язык программирования Python”.

|  |  |  |
| --- | --- | --- |
| **Студент** группа ИДБ-20-08 |  | Валеев Г. А. |
|  | подпись |  |
| Руководитель  старший преподаватель |  | Сакович С. |
|  | подпись |  |

Москва 2022 г.

**ОГЛАВЛЕНИЕ**

[ГЛАВА 1. выполнение лабораторной работы 3](#_Toc85480324)

[1.1. Задание №1 3](#_Toc85480325)

[1.2. Задание №2 3](#_Toc85480326)

[1.3. Задание №3 4](#_Toc85480327)

[1.4. Задание №4 5](#_Toc85480328)

[1.5. Задание №5 6](#_Toc85480328)

[ВЫВОД 8](#_Toc85480331)

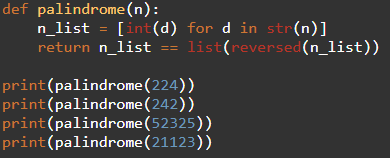
1. выполнение лабораторной работы

Цель лабораторной работы: написать программный код и реализовать алгоритмы для решения всех поставленных задач на определенный балл. В процессе лабораторной работы были выполнены все задания на максимальный балл. Программный код реализации всех заданий представлен в GitHub-репозитории: <https://github.com/Shamopol/L2>

# Задание №1

Задание: написать функцию, которая принимает int и возвращает true/false в зависимости от того, является ли число палиндромом или нет.

Палиндром – число, которое в определённой позиционной системе исчисления читаются одинаково как справа налево, так и слева направо. Таким образом, для проверки, является ли число палиндромом или нет, необходимо перевернуть число (расставить цифры задом наперед) и сравнить его с исходным числом. Программная реализация (рис. 1.1.1) и результаты выполнения функции (рис. 1.1.2) представлены ниже.



* + 1. Программная реализация функции для проверки палиндрома



* + 1. Результат работы функции для проверки палиндрома

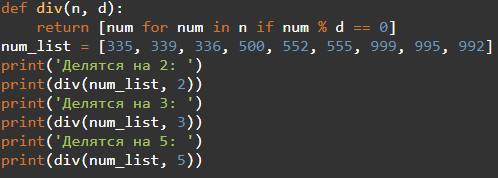
# Задание №2

Задание: написать функцию, которая принимает на вход список положительных из числовых элементов и возвращает 3 списка:

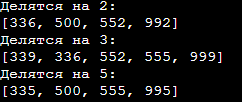
1. числа, делящиеся на 2;
2. числа, делящиеся на 3;
3. числа, делящиеся на 5;

Для проверки делится ли число на другое без остатка используется оператор взятия остатка при делении «%». Если остаток от деления равен нулю, то данное число является делителем исходного числа.

Реализация функции принимает список чисел и делитель. Возвращает список всех чисел, которые делятся на делитель без остатка. Программная реализация (рис. 1.2.1) и результаты выполнения функции (рис. 1.2.2) представлены ниже.



* + 1. Реализация простой функции нахождения чисел,  
       делящихся без остатка



* + 1. Результат простой функции нахождения чисел,  
       делящихся без остатка

# Задание №3

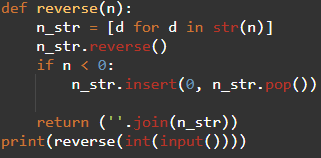
Задание: написать функцию, принимающую на вход int, выдающую int, обратное входному значению. Например:

Input: -123

Output: -321

Стандартный случай получения числа, обратного исходному, предполагает простую конвертацию числа в список цифр и его разворот. Однако, в задании дается отрицательное число, где минус должен остаться на первом месте. Так же отрицательно число нельзя конвертировать в список цифр типа int, так как минус в отдельности не является целочисленным типом. Поэтому следует работать через список строк.

Сначала необходимо преобразовать число в строку и разбить её на отдельные элементы, образовав таким образом список символов цифр. Далее производится разворот списка, чтобы все элементы списка шли в обратном порядке. Однако, если число отрицательное, то минус будет занимать последнее места в таком списке. Следовательно, необходимо убрать минус с последнего места в списке и расположить его вначале. Далее производится конвертация списка строк в одну строку и её дальнейшее преобразование в тип int. Программная реализация (рис. 1.3.1) и результаты выполнения функции (рис. 1.3.2) представлены ниже.



* + 1. Программная реализация функции разворота числа



* + 1. Результат работы функции разворота числа

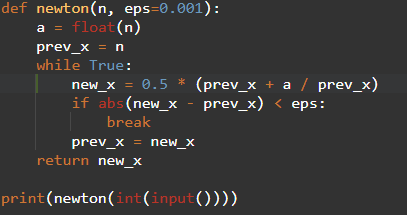
# Задание №4

Задание: написать функцию, которая будет рассчитывать квадратный корень методом Ньютона.

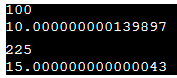
Метод Ньютона заключается в нахождении последовательного приближения числа по следующей формуле:

Нахождение квадратного является частным случаем метода Ньютона для функции . Подставив данное равенство в основную форму, упростим выражение и получим:

Составим программную реализацию метода Ньютона по данной формуле. Также ограничим точность числа нахождения числа через . Программная реализация (рис. 1.4.1) и результаты выполнения функции (рис. 1.4.2) представлены ниже.



* + 1. Программная реализация функции нахождения квадратного корня методом Ньютона

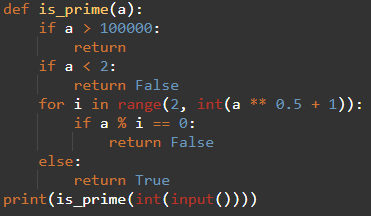


* + 1. Результат работы функции нахождения квадратного корня методом Ньютона

# Задание №5

Задание: написать функцию, принимающую 1 аргумент — число от 0 до 100000, и возвращающую true, если оно простое, false если нет.

Программная реализация (рис. 1.5.1) и результаты выполнения функции (рис. 1.5.2) представлены ниже.



* + 1. Программная реализация функции определения простых чисел



* + 1. Результат работы функции определения простых чисел

ВЫВОД

В процессе выполнения лабораторной работы закрепил знание основных алгоритмов, а также полностью освоил базовый синтаксис языка программирования Python. Вспомнил метод нахождения квадратного корня Ньютона и детально изучил процесс написания декораторов, а также передачу им параметров.