|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** – **Российский технологический университет»**  **РТУ МИРЭА** | | | | |
| Институт информационных технологий (ИТ) | |
| Кафедра прикладной математики | |

|  |  |
| --- | --- |
| **ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1** | |
| **по дисциплине «Технологии и инструментарий анализа больших данных»** | |
|  | |
| Выполнил студент группы ИКБО-20-21 | Сидоров С.Д. |
| Проверил ассистент кафедры ПМ ИИТ | Тетерин Н.Н. |

**Задание 1**

Установить Python, если это не было сделано ранее.

**Задание 2**

Написать программу, которая вычисляет площадь фигуры, параметры которой подаются на вход. Фигуры, которые подаются на вход: треугольник, прямоугольник, круг. Результатом работы является словарь, где ключ – это название фигуры, а значение – это площадь.

Код к заданию 2 представлен на рисунке 1.



Рисунок 1 – Код к заданию 1

Пример выполнения представлен на рисунке 2.

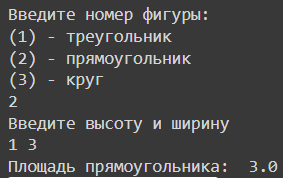


Рисунок 2 – Пример работы

**Задание 3**

Написать программу, которая на вход получает два числа и операцию, которую к ним нужно применить. Должны быть реализованы следующие операции: +, -, /, //, abs – модуль, pow или \*\* – возведение в степень.

Код к заданию 3 представлен на рисунке 3.

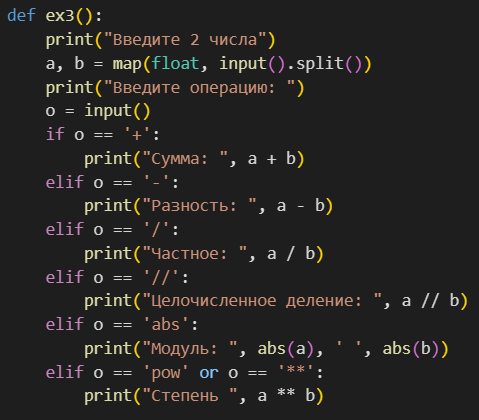


Рисунок 3 – Код к заданию 3

Пример работы представлен на рисунке 4.

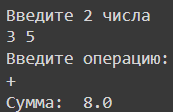


Рисунок 4 – Пример работы

**Задание 4**

Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и после этого выводит сумму квадратов всех считанных чисел.

Код к заданию 4 представлен на рисунке 5.

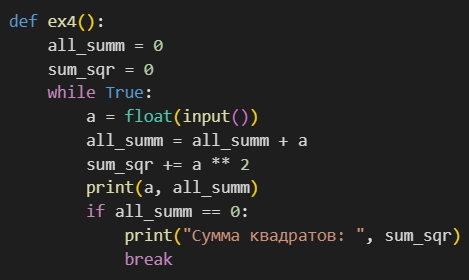


Рисунок 5 – Код к заданию 4

Пример работы представлен на рисунке 6.

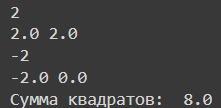


Рисунок 6 – Пример работы

**Задание 5**

Напишите программу, которая выводит последовательность чисел, длинною N, где каждое число повторяется столько раз, чему оно равно. На вход программе передаётся неотрицательное целое число N. Например, если N = 7, то программа должна вывести 1 2 2 3 3 3 4. Вывод элементов списка через пробел – print(\*list).

Код к заданию 5 представлен на рисунке 7.

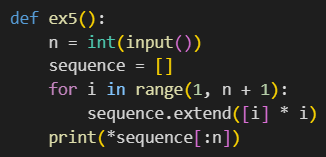


Рисунок 7 – Код к заданию 5

Пример работы представлен на рисунке 8.



Рисунок 8 – Пример работы

**Задание 6**

А = [1, 2, 3, 4, 2, 1, 3, 4, 5, 6, 5, 4, 3, 2] В = [‘a’, ’b’, ’c’, ’c’, ’c’, ’b’, ’a’, ’c’, ’a’, ’a’, ’b’, ’c’, ’b’, ’a’] Создать словарь, в котором ключи – это содержимое списка В, а значения для ключей словаря – это сумма всех элементов списка А в соответствии с буквой, содержащийся на той же позиции в списке В. Пример результата программы: {‘a’ : 10, ‘b’ : 15, ‘c’ : 6}.

Код к заданию 6 представлен на рисунке 9.

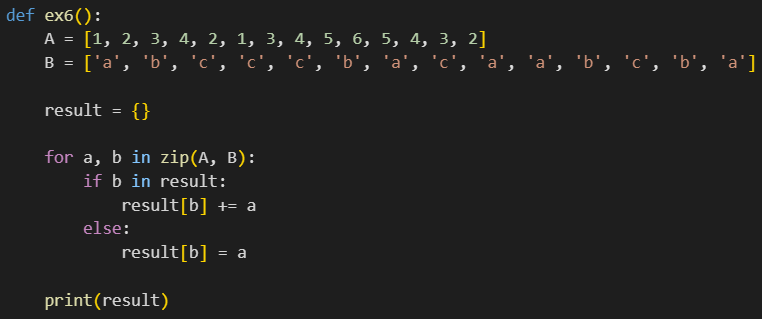


Рисунок 9 – Код к заданию 6

Пример работы представлен на рисунке 10.



Рисунок 10 – Пример работы

**Задание 7**

Скачать и загрузить данные о стоимости домов в калифорнии, используя библиотеку sklearn.

**Задание 8**

Использовать метод info().

**Задание 9**

Узнать, есть ли пропущенные значения, используя isna().sum().

**Задание 10**

Вывести записи, где средний возраст домов в районе более 50 лет и население более 2500 человек, используя метод loc().

**Задание 11**

Узнать максимальное и минимальное значения медианной стоимости дома.

**Задание 12**

Используя метод apply(), вывести на экран название признака и его среднее значение.

Код к заданиям 7 – 12 представлен на рисунке 11.

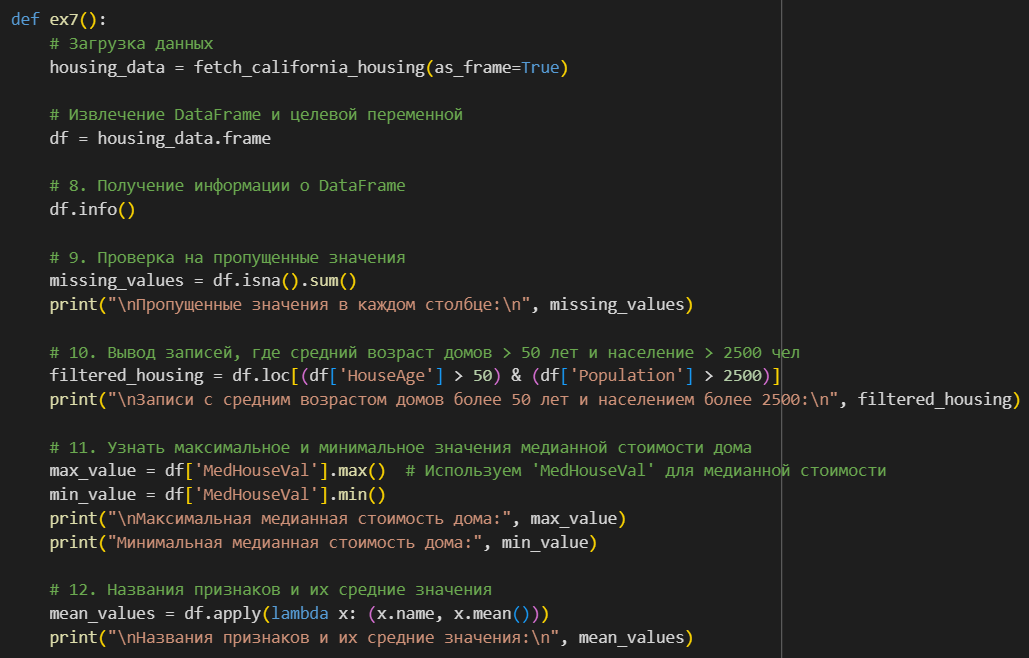


Рисунок 11 – Код к заданиям 7-12

Пример работы представлен на рисунках 12, 13.

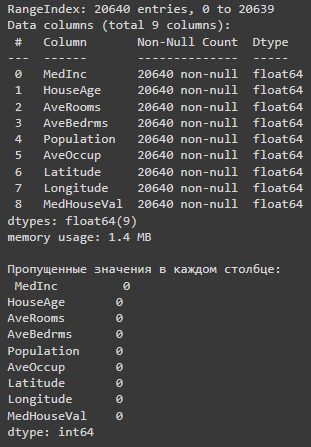


Рисунок 12 – Пример работы

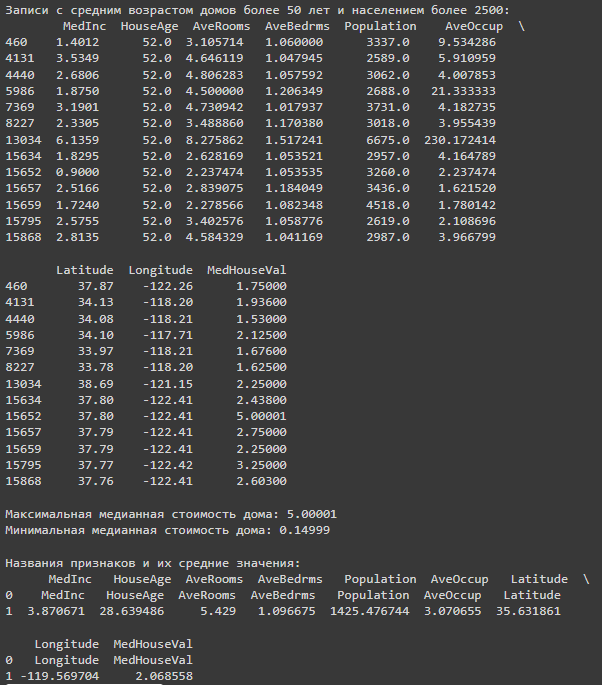


Рисунок 13 – Пример работы

**Задание 1\***

Дан текст на английском языке. Необходимо закодировать его с помощью азбуки Морзе, где каждой букве соответствует последовательность точек и тире. Например, буква «g» превратится в строку «--.». В переменной morze для удобства хранится словарь соответствия латинских букв коду Морзе. morze = {'a': '.-', 'b': '-…', 'c': '-.-.', 'd': '-..', 'e': '.', 'f': '..-.', 'g': '--.', 'h': '….', 'i': '..', 'j': '.---', 'k': '-.-', 'l': '.-..', 'm': '--', 'n': '-.', 'o': '---', 'p': '.--.', 'q': '--.-', 'r': '.-.', 's': '…', 't': '-', 'u': '..-', 'v': '…-', 'w': '.--', 'x': '-..-', 'y': '-.--', 'z': '--..'}

На входе: В одной строке вам дан текст, который состоит из латинских букв и пробелов.

На выходе: Выведите каждое слово исходного текста, закодированное азбукой Морзе. Количество строк в ответе должно совпадать с количеством слов в исходном тексте. Между закодированными буквами ставится ровно один пробел. Например, слово «Help» превратится в «.... . .-.. .--.». Строчные и заглавные буквы кодируются одинаково.

Код программы представлен на рисунке 14.



Рисунок 14 – Код к заданию 1\*

Пример работы представлен на рисунке 15.

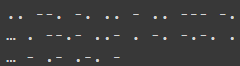


Рисунок 15 – Пример работы

**Задание 2\***

В некотором городе открывается новая служба по доставке электронных писем. Необходимо наладить систему регистрации новых пользователей. Регистрация должна работать следующим образом: если новый пользователь хочет зарегистрироваться на сайте, то он должен послать системе запрос name со своим именем. Система должна определить, существует ли уже такое имя в базе данных. Если такого имени не существует, то оно заносится в базу данных системы и пользователю возвращается ответ "ОК", подтверждающий успешную регистрацию. А если пользователь с таким именем уже существует, то система должна сформировать новое имя и выдать его пользователю в качестве подсказки, при этом сама подсказка также добавляется в базу данных. Новое имя формируется следующим образом: к name последовательно приписываются числа, начиная с 1 (name1, name2 и так далее), и среди них находят такое наименьшее i, что namei еще не содержится в системе.

Входные данные В первой строке входных данных задано число n (1 ≤ n ≤ 100000). Следующие n строк содержат запросы к системе. Каждый запрос представляет собой непустую строку длиной не более 32 символов, состоящую только из строчных букв латинского алфавита.

Выходные данные В выходных данных должно содержаться n строк – ответы системы на запросы: "OK" в случае успешной регистрации, или подсказка с новым именем, если запрашиваемое уже занято

Код к заданию 2\* представлен на рисунке 16.

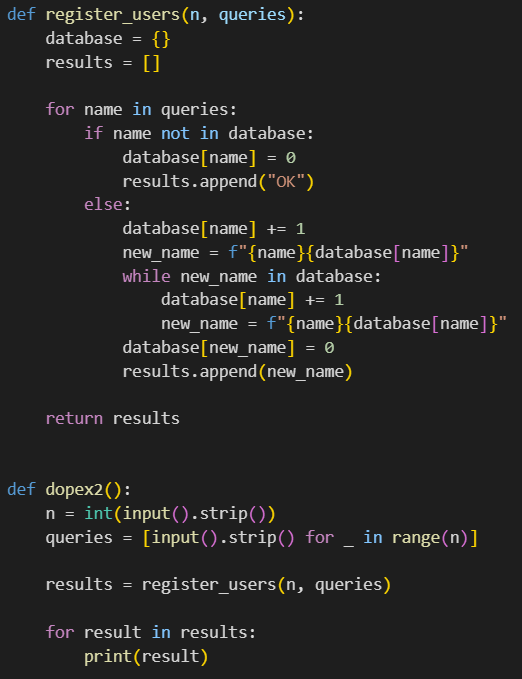


Рисунок 16 – Код к заданию 2\*

Пример работы представлен на рисунке 17.

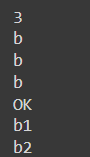


Рисунок 17 – Пример работы

**Задание 3\***

Необходимо создать программу обработки запросов пользователей к файловой системе компьютера. Над каждым файлом можно производить следующие действия: запись – w ("write"), чтение – r ("read"), запуск – x ("execute").

Входные данные На вход программе подаются следующие параметры: число n – количество файлов в файловой системе. В следующих n строках содержится информация с именами файлов и допустимыми действиями (w, x, r), разделенных пробелами. Далее идет число m – количество запросов к файлам вида «операция файл» (обозначение операции: "write", "read", "execute").

Выходные данные Для каждого допустимого запроса программа должна возвращать OK, для недопустимого – Access denied.

Код к заданию 3\* представлен на рисунке 18.

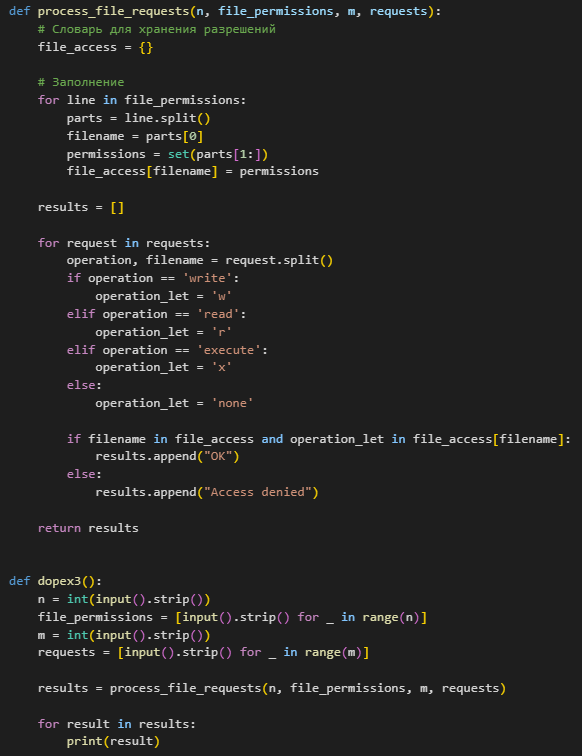


Рисунок 18 – Код к заданию 3\*

Пример работы представлен на рисунке 19.

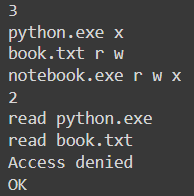


Рисунок 19 – Пример работы