**Билет 1**

**1.** (Задание на **D**) Написать функцию, которая принимает на вход строку, содержащую русское слово, и целое число, выражающее номер ударного слога в этом слове, и возвращающую строку, содержащую это же слово с цифрой 1 после ударного гласного.

def place\_stress(word: str, stressed\_syl: int) -> str:

 ...

>>>place\_stress("мама", 1)

'ма1ма'

**2a.** (Задание на **C**) Написать программу, использующую функцию из предыдущего задания. Программа должна открыть текстовый файл, в котором на каждой строке находится слово и номер ударного слога в нём, разделённые пробелом, считать эту информацию, расставить ударения в словах и записать их в новый файл в отсортированном порядке, каждое слово на своей строке.

Например:

Входной файл:

папа 1

мама 1

Выходной файл:

ма1ма

па1па

**2b.** (Задание на **B**) Задание, аналогичное заданию 2a, но программа должна записывать слова в два файла: в один - слова, состоящие из двух слогов, в другой - все остальные.

Оценка **A** ставится в случае корректного выполнения задания 2b при условии, что код оформлен аккуратно и вызывает минимум замечаний.

**Билет 2**

**1.** (Задание на **D**) Написать функцию, которая принимает на вход список кортежей, каждый из которых содержит два вещественных числа, и возвращающую сумму расстояний между последовательными точками, координаты которых выражены этими числами. Для вычисления расстояния используйте функцию, вычисляющую квадратный корень:

>>>from math import sqrt

>>>sqrt(2)

1.4142135623730951

def calculate\_distance(points: list[tuple[float]]) -> float:

 ...

>>>calculate\_distance([(0.0, 0.0), (1.0, 0.0), (1.0, 3.5)])

4.5

**2a.** (Задание на **C**) Написать программу, использующую функцию из предыдущего задания. Программа должна открыть текстовый файл, в котором на каждой строке находится идентификатор последовательности точек и сама последовательность (точки разделены пробелами, координаты точек - запятыми), считать эту информацию и записать результат обработки в новый файл, в котором на каждой строке будет идентификатор последовательности и её суммарное расстояние (по возрастанию расстояния).

Например:

Входной файл:

D01 0.0,0.0 1.0,0.0 1.0,3.5

D02 0.0,1.0 1.0,0.0

Выходной файл:

D02 1.4142135623730951

D01 4.5

**2b.** (Задание на **B**) Задание, аналогичное заданию 2a, но программа должна записывать расстояния в два файла: в один - расстояния, меньшие определённого значения X, в другой - большие. Значение X вычисляется как среднее арифметическое максимального и минимального найденного расстояния.

Оценка **A** ставится в случае корректного выполнения задания 2b при условии, что код оформлен аккуратно и вызывает минимум замечаний.

**Билет 3.**

**1.** (Задание на **D**) Написать функцию, которая принимает на вход строку, содержащую русское предложение, и возвращающую строку, в которой каждая буква заменена её порядковым номером в алфавите, соседние номера разделены дефисами. Пробелы и другие символы заменяются цифрой 0.

def num\_code(message: str) -> str:

 ...

>>>num\_code("мама мыла раму")

'14-1-14-1-0-14-29-13-1-0-18-1-14-21'

**2a.** (Задание на **C**) Написать программу, использующую функцию из предыдущего задания. Программа должна открыть текстовый файл, в котором на каждой строке находится сообщение, считать эту информацию и записать результат шифрования в новый файл в отсортированном порядке, каждое сообщение на своей строке.

Например:

Входной файл:

был тихий серый вечер

мама мыла раму

Выходной файл:

14-1-14-1-0-14-29-13-1-0-18-1-14-21

2-29-13 20-10-23-10-11 19-6-18-29-11 3-6-25-6-18

**2b.** (Задание на **B**) Задание, аналогичное заданию 2a, но программа должна записывать результат в два файла: в один - сообщения, сумма чисел в которых чётная, в другой - сообщения, сумма чисел в которых нечётная.

Оценка **A** ставится в случае корректного выполнения задания 2b при условии, что код оформлен аккуратно и вызывает минимум замечаний.