Лабораторная работа №4

Для выполнения данной лабораторной работы Вам необходимо выполнить все **три** задания, представленные ниже.

Все три задания должны быть помещены в репозиторий «*operators\_lz*» в GitHUB. Каждое задание выполняется в отдельном файле, должно быть снабжено комментариями, а также иметь дружественный интерфейс (пользователь должен понимать, что вводит, и что получает назад).

Входные данные могут быть получены как из файла, так и с клавиатуры по усмотрению выполняющего работу. Выходные данные должны иметь такой же вид как и входные: если ввод происходит из файла, то вывод делается в файл, если ввод с клавиатуры, то вывод – в консоль.

**Задание 1.**

Создайте файл «*fibonacci.py*».

В файле реализуйте программный код, который будет выводить ряд Фибоначчи до элемента, порядковый номер которого указан в исходных данных.

Исходные данные имеют вид одной строки, где по порядку указаны значения первого элемента прогрессии ***b***, множитель прогрессии ***q*** и номер последнего элемента ***n***.

Выведите пользователю результат вычисления суммы геометрической прогрессии.

Входные данные: -10\_000 ≤ ***b*** ≤ 10\_000, 1 ≤ ***q*** ≤ 50, 2 ≤ ***n*** ≤ 100.

Пример:

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5 | 0, 1, 1, 2, 3, 5 |
| 20 | 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765 |

**Задание 2.**

Создайте файл «*simple.py*».

В файле реализуйте программный код, который будет определять простое ли число.

Исходные данные имеют вид одной строки, где указано всего одно число ***n***. Определите, является ли ***n*** простым числом.

Выведите пользователю результат определения простоты числа.

Входные данные: 1 ≤ ***n*** ≤ 25\_000.

Пример:

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 21434 | N |
| 21467 | Y |

**Задание 3.**

Создайте файл «*sum of series.py*».

Написать программу на Python, которая считает количество итераций, необходимых для того, чтобы сумма членов ряда приблизилась к заданному числу с точностью ε. Пользователь вводит два значения:

* Стартовое значение ряда (начальная сумма).
* Конечное значение, к которому стремится ряд с точностью **ε**.

Условия:

* Программа должна принимать от пользователя стартовое и конечное значения, а также желаемую точность **ε**.
* Программа должна вычислять, сколько итераций нужно, чтобы сумма элементов ряда достигла или превысила конечное значение с точностью **ε**.

Исходные данные имеют вид одной строки, где указаны число ***L*** – начальная сумма, число **ε** – количество знаков после запятой (точность) и искомая сумма ряда ***S***.