САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №1

по курсу «Алгоритмы и структуры данных»

Тема: Тема работы

Выполнила:

Коновалова Кира Романовна

К3139

Проверил:

Афанасьев А. В.

Санкт-Петербург

2024г.

# Содержание отчета

[Содержание отчета 2](#_Toc177655099)

[Задачи по варианту 3](#_Toc177655100)

[**Задача №1. Ввод-вывод Подпункт №1** 3](#_Toc177655101)

[**Задача №1. Ввод-вывод Подпункт №2** 4](#_Toc177655102)

[**Задача №1. Ввод-вывод Подпункт №3** 5](#_Toc177655103)

[**Задача №1. Ввод-вывод Подпункт №4** 6](#_Toc177655104)

[**Задача №2. Число Фибоначчи** 8](#_Toc177655105)

[**Задание №3. Еще про числа Фибоначчи** 11](#_Toc177655106)

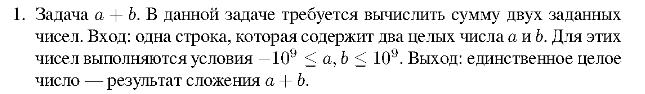
[**Задача №4. Тестирование ваших алгоритмов** 13](#_Toc177655107)

[Вывод: 16](#_Toc177655108)

# Задачи по варианту

**Задача №1. Ввод-вывод Подпункт №1**

**Текст задачи:**



**Код:**

a, b = map(int, input().split())

f = True

*while* f:

*if* (-10 \*\* 9 <= a <= 10 \*\* 9) and (-10 \*\* 9 <= b <= 10 \*\* 9):

        print(a + b)

*break*

*else*:

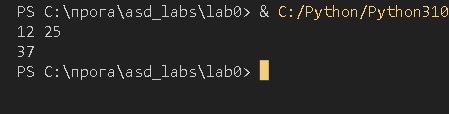
        print('Числа не подходят по диапазону')

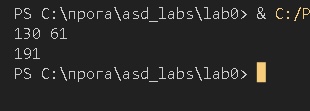
        a, b = map(int, input().split())

**Текстовое объяснение задачи:**

В переменные a и b считываются значения, которые вводятся с клавиатуры, и затем, если введенные числа удовлетворяют условию -10^9<=a, b<=10^9, то на экран выводиться значение a + b

**Пример работы кода:**



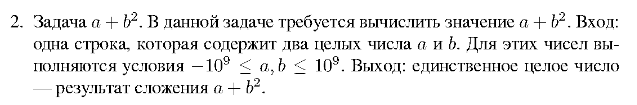


**Вывод по задаче:**

Программа принимает два целых числа, вычисляет их сумму и выводит результат. Код выполняет поставленную задачу.

**Задача №1. Ввод-вывод Подпункт №2**

**Текст задачи:**



**Код:**

a, b = map(int, input().split())

f = True

*while* f:

*if* (-10 \*\* 9 <= a <= 10 \*\* 9) and (-10 \*\* 9 <= b <= 10 \*\* 9):

        print(a + b\*\*2)

*break*

*else*:

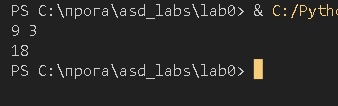
        print('Числа не подходят по диапазону')

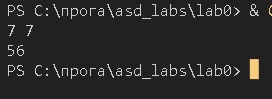
        a, b = map(int, input().split())

**Текстовое объяснение задачи:**

В переменные a и b считываются значения, которые вводятся с клавиатуры, и затем, если введенные числа удовлетворяют условию -10^9<=a, b<=10^9, то на экран выводиться значение a + b^2

**Пример работы кода:**



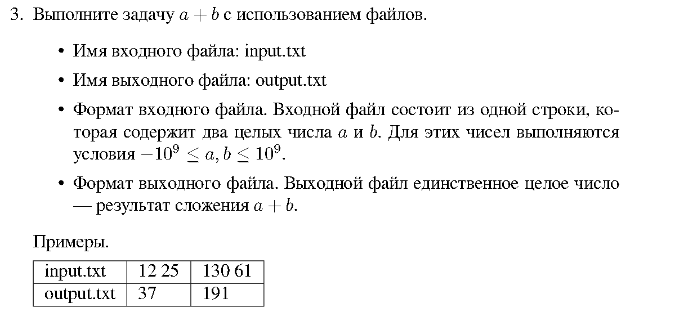


**Вывод по задаче:**

Программа принимает два целых числа, вычисляет сумму a + b^2 и выводит результат. Код выполняет поставленную задачу.

**Задача №1. Ввод-вывод Подпункт №3**

**Текст задачи:**



**Код:**

*with* open("input.txt", "r") *as* f:

    a, b = map(int, f.readline().split())

f.close()

*with* open("output.txt", "w") *as* f:

*if* (-10\*\*9 <= a <= 10\*\*9) and (-10\*\*9 <= b <= 10\*\*9):

        f.write(str(a + b))

*else*:

        print('Числа не подходят по диапазону')

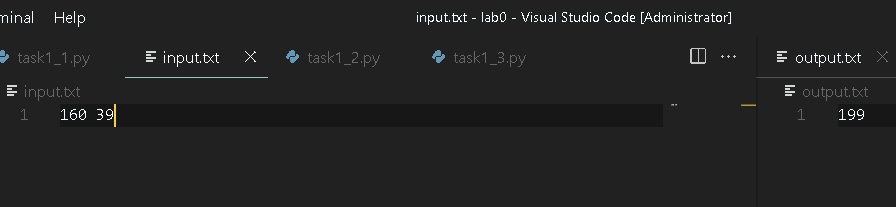
f.close()

**Текстовое объяснение задачи:**

1. Открывается файл input.txt с входными данными
2. Считывается первая строка входного файла. Из него берутся числа a и b. Входной файл закрывается.
3. Открывается выходной файл output.txt для записи. Числа из входного файла проверяются на соответствие интервалу, если условие выполняется, то в выходной файл записывается сумма a и b, иначе выводиться ошибка. Файл закрывается.

**Пример работы кода:**



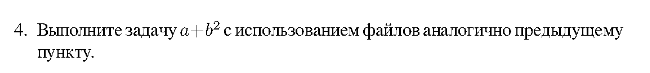


**Вывод по задаче:**

Данная задача подобна первому подпункту задания 1, но требует умения работать с файлами. Выполнение этой задачи позволило мне отработать выполнение файлового вводы и вывода в Python (чтение из входного файла, и запись в другой выходной файл).

**Задача №1. Ввод-вывод Подпункт №4**

**Текст задачи:**



**Код:**

*with* open("input.txt", "r") *as* f:

    a, b = map(int, f.readline().split())

f.close()

*with* open("output.txt", "w") *as* f:

*if* (-10\*\*9 <= a <= 10\*\*9) and (-10\*\*9 <= b <= 10\*\*9):

        f.write(str(a + b \*\* 2))

*else*:

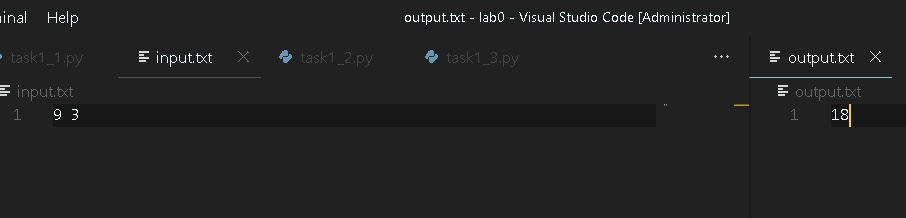
        print('Числа не подходят по диапазону')

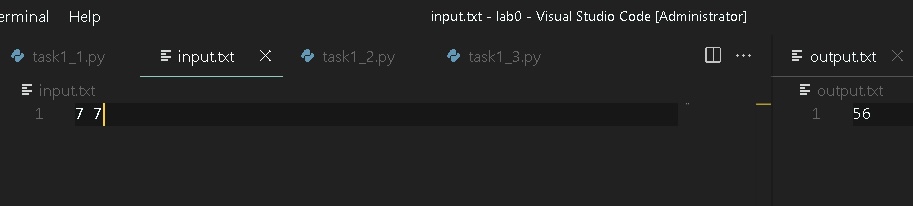
f.close()

**Текстовое объяснение задачи:**

1. Открывается файл input.txt с входными данными
2. Считывается первая строка входного файла. Из него берутся числа a и b. Входной файл закрывается.
3. Открывается выходной файл output.txt для записи. Числа из входного файла проверяются на соответствие интервалу, если условие выполняется, то в выходной файл записывается сумма a и b^2, иначе выводиться ошибка. Файл закрывается.

**Пример работы кода:**



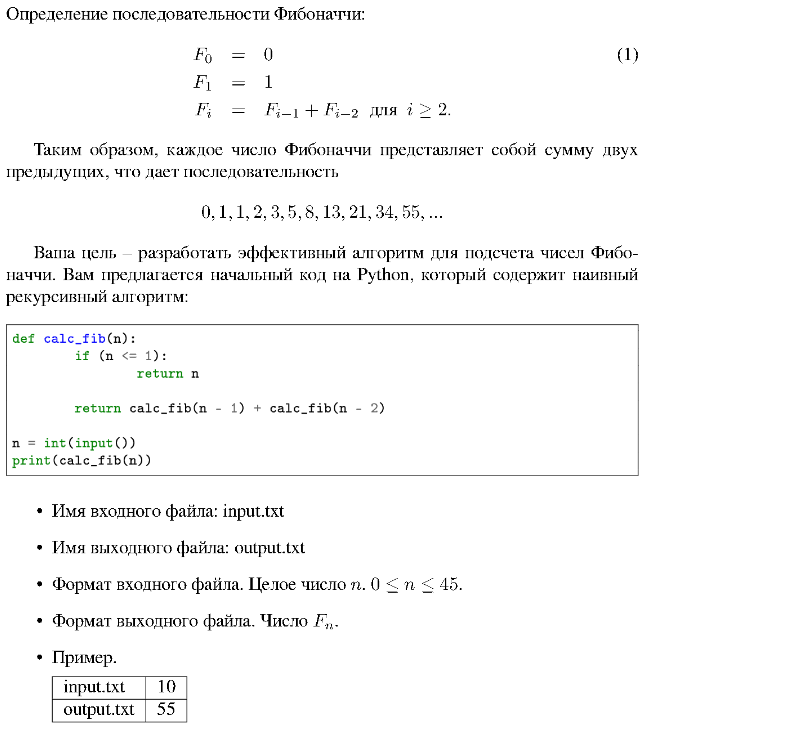


**Вывод по задаче:**

Данная задача подобна подпункту два задания 1, но требует умения работать с файлами. Задача имеет чтение и запись в файл и позволила закрепить материал по работе с файлами в Python

**Задача №2. Число Фибоначчи**

**Текст задачи:**

****

**Код:**

*with* open("input.txt", "r") *as* f:

    n = int(f.readline())

f.close()

*with* open("output.txt", "w") *as* f:

*if* 0 <= n <= 45:

*if* n <= 1:

            f.write(str(n))

*else*:

            a = [0, 1]

*for* i *in* range(1, n):

                a.append((a[i-1] + a[i]))

            print(\*a)

            f.write(str(a[n]))

*else*:

        print('Числа не подходят по диапазону')

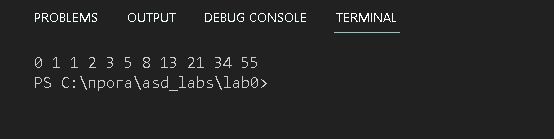
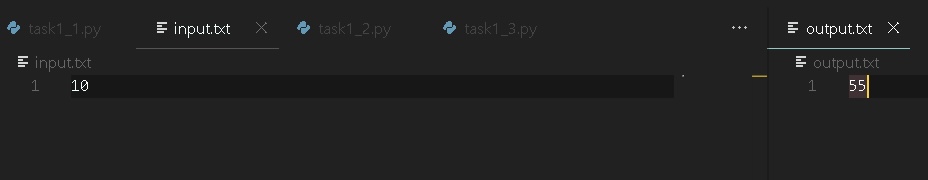
        f.write('out of range')

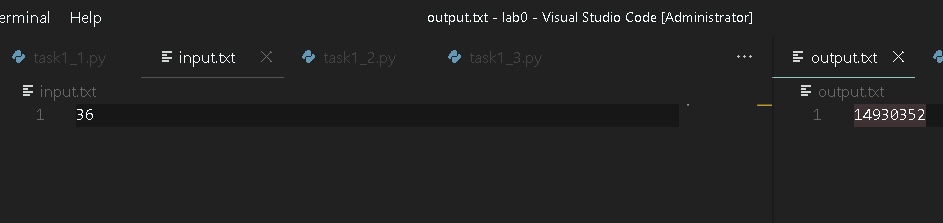
f.close()

**Текстовое объяснение задачи:**

1. Открываем файл input.txt для чтения входных данных. Считываем данные n и закрываем файл.
2. Открываем выходной файл output.txt для записи.
3. Делаем проверку числа на принадлежность к диапазону и, если число подходит, выполняем алгоритм:
   1. Если входное число 0 или 1, то выводим соответственно 0 и 1
   2. Иначе запускаем цикл, в котором находим сумму двух предыдущих чисел
   3. Выводим для проверки все числа Фибоначчи
   4. в выходной файл записываем n-ое число Фибоначчи
4. Если число не принадлежит нужному нам диапазону – выводим ошибку
5. Закрываем выходной файл

**Пример работы кода:**

****

****

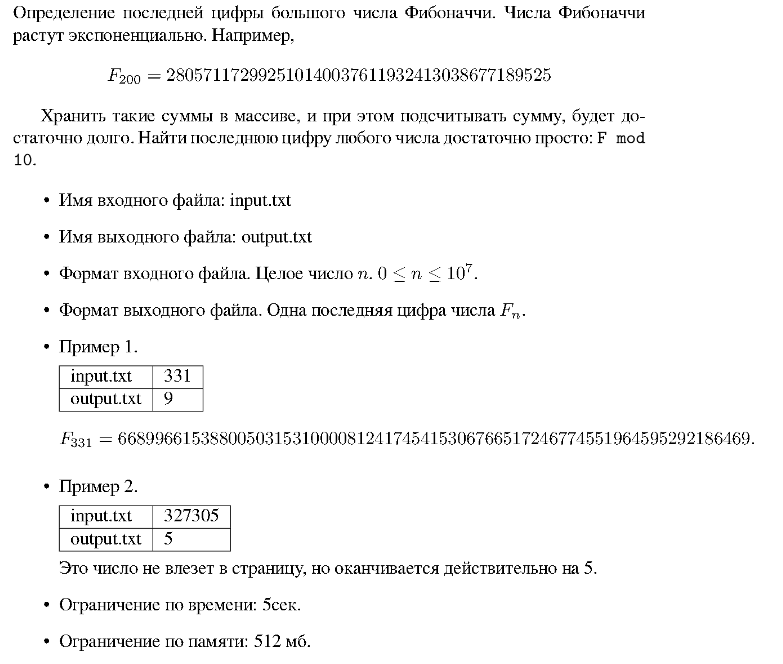
****

**Вывод по задаче:**

При решении задачи потребовались знания по работе с файлами и написать алгоритм для подсчета числа Фибоначчи. Код выполняет поставленную задачу.

**Задание №3. Еще про числа Фибоначчи**

**Текст задачи:**

****

**Код:**

*with* open("input.txt", 'r') *as* f:

    n = int(f.readline())

f.close()

*with* open("output.txt", "w") *as* f:

*if* 0 <= n <= 10\*\*7:

*if* n <= 1:

            f.write(str(n))

*else*:

            a, b = 0, 1

            s = 0

*for* i *in* range(1, n):

                s = (a+b)%10

                a, b = b, s

            f.write(str(s))

*else*:

        print('Числа не подходят по диапазону')

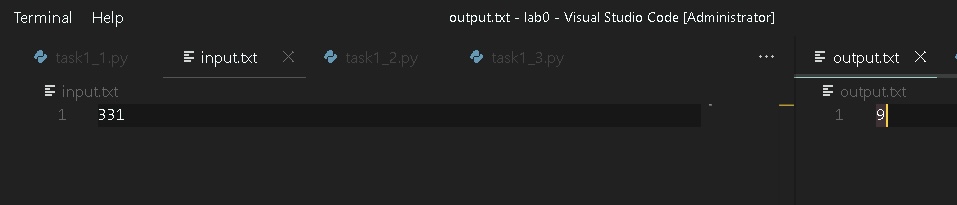
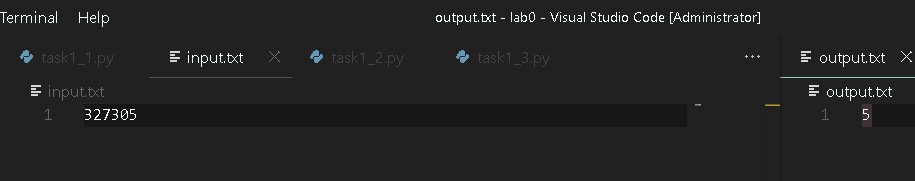
        f.write('out of range')

f.close()

**Текстовое объяснение задачи:**

1. Открываем файл input.txt для чтения входных данных. Считываем данные n и закрываем файл.
2. Открываем выходной файл output.txt для записи.
3. Делаем проверку числа на принадлежность к диапазону и, если число подходит, выполняем алгоритм:
   1. Если входное число 0 или 1, то выводим соответственно 0 и 1
   2. Иначе запускаем цикл, в котором создаем переменные a и b, содержащие 1 и 2 элемент Фибоначчи. Запускаем цикл, который находит последнию цифру числа. Далее записываем в переменную a значение переменной b, а в b записываем значение s. Далее мы будем работать только с последними цифрами чисел.
   3. в выходной файл записываем результат s
4. Если число не принадлежит нужному нам диапазону – выводим ошибку
5. Закрываем выходной файл

**Пример работы кода:**

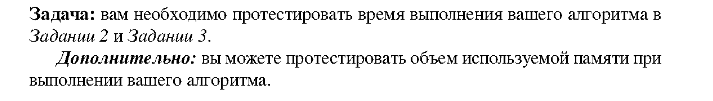
****

**Вывод по задаче:**

Для решения данной задачи понадобилось немного изменить задачу номер 3, применить знания по работе с файлами и алгоритм вычисления чисел Фибоначчи

**Задача №4. Тестирование ваших алгоритмов**

**Текст задачи:**

****

**Тестирование второй задачи.**

**Код:**

*import* time

st\_time = time.perf\_counter()

*with* open("input.txt", "r") *as* f:

    n = int(f.readline())

f.close()

*with* open("output.txt", "w") *as* f:

*if* 0 <= n <= 45:

*if* n <= 1:

            f.write(str(n))

*else*:

            a = [0, 1]

*for* i *in* range(1, n):

                a.append((a[i-1] + a[i]))

            print(\*a)

            f.write(str(a[n]))

*else*:

        print('Числа не подходят по диапазону')

        f.write('out of range')

f.close()

print("Время работы: %s секунд " % (time.perf\_counter() - st\_time))

**Текстовое объяснение задачи:**

1. Импортируем модуль time, который необходим для отслеживания времени работы алгоритма
2. Записываем задачу номер два (код описан в соответствующей главе)
3. Выводим время выполнения алгоритма

|  |  |
| --- | --- |
|  | Время выполнения |
| Нижняя граница диапазона значений входных данных из текста задачи | Время работы: 0.000514900078997016 секунд |
| Пример из задачи | Время работы: 0.00070830003824085 секунд |
| Верхняя граница диапазона значений входных данных из текста задачи | Время работы: 0.000865599955432117 секунд |

**Вывод по задаче:**

Скорость выполнения алгоритма растет линейно: чем больше мы подаём на вход число n, тем дольше выполняется программа

**Тестирование третей задачи.**

**Код:**

*import* time

st\_time = time.perf\_counter()

*with* open("input.txt", 'r') *as* f:

    n = int(f.readline())

f.close()

*with* open("output.txt", "w") *as* f:

*if* 0 <= n <= 10\*\*7:

*if* n <= 1:

            f.write(str(n))

*else*:

            a, b = 0, 1

            s = 0

*for* i *in* range(1, n):

                s = (a+b)%10

                a, b = b, s

            f.write(str(s))

*else*:

        print('Числа не подходят по диапазону')

        f.write('out of range')

f.close()

print("Время работы: %s секунд " % (time.perf\_counter() - st\_time))

**Текстовое объяснение задачи:**

1. Импортируем модуль time, который необходим для отслеживания времени работы алгоритма
2. Записываем задачу номер три (код описан в соответствующей главе)
3. Выводим время выполнения алгоритма

|  |  |
| --- | --- |
|  | Время выполнения |
| Нижняя граница диапазона значений входных данных из текста задачи | Время работы: 0.0005406000418588519 секунд |
| Пример из задачи | Время работы: 0.0006834000814706087 секунд |
| Пример из задачи | Время работы: 0.031066900002770126 секунд |
| Верхняя граница диапазона значений входных данных из текста задачи | Время работы: 1.006157899973914 секунд |

**Вывод по задаче:**

Скорость выполнения алгоритма растет линейно: чем больше мы подаём на вход число n, тем дольше выполняется программа

**Вывод:**

В данной лабораторной работе мы научились работать со сторонними файлами, научились измерять время работы различных алгоритмов, благодаря чему смогли наглядно увидеть линейную зависимость времени работы алгоритма от входных данных