

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №0
по курсу «Алгоритмы и структуры данных»

Тема: Введение

Вариант 1

Выполнил:

Журбина Марина Андреевна

Группа К3139

Проверил:

Афанасьев А. В.

Санкт-Петербург

2024 г.

Содержание отчета

Оглавление

<i>Содержание отчета</i>	<i>2</i>
<i>Задачи по варианту</i>	<i>3</i>
Задание № 1.1. $a+b$	3
Задание №1.2. $a+b^2$	4
Задание №1.3. $a+b$ с использованием файлов	4
Задание №1.4. $a+b^2$ с использованием файлов	6

Задачи по варианту

Задание № 1.1. $a+b$

Текст задачи:

1. Задача $a + b$. В данной задаче требуется вычислить сумму двух заданных чисел. Вход: одна строка, которая содержит два целых числа a и b . Для этих чисел выполняются условия $-10^9 \leq a, b \leq 10^9$. Выход: единственное целое число — результат сложения $a + b$.

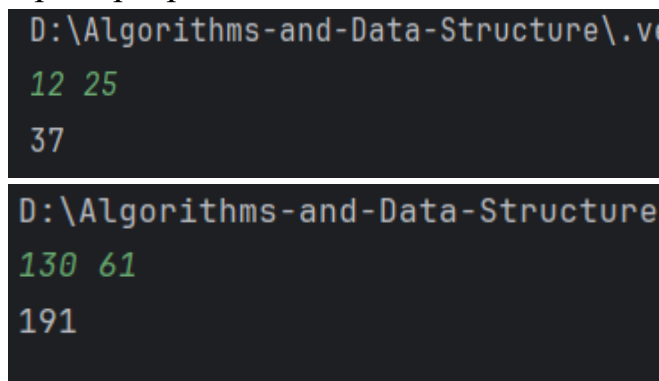
Код программы:

```
_min = 10 ** (-9)
_max = 10 ** 9
a, b = map(int, input().split())
if (_min <= a <= _max) and (_min <= b <= _max):
    print(a + b)
else:
    print('Введены некорректные данные')
```

Текстовое объяснение решения:

- 1) Заданы левая и правая границы интервала допустимых значений.
- 2) Считываются два числа, введенные пользователем.
- 3) Числа проверяются на соответствие интервалу, при выполненном условии выводится ответ на задачу.

Примеры работы кода:



```
D:\Algorithms-and-Data-Structure\.v
12 25
37

D:\Algorithms-and-Data-Structure
130 61
191
```

Вывод по задаче:

В решении задачи использованы переменные, операторы сравнения и if-else конструкции. Код выполняет поставленную задачу.

Задание №1.2. $a+b^2$

Текст задачи:

2. Задача $a + b^2$. В данной задаче требуется вычислить значение $a + b^2$. Вход: одна строка, которая содержит два целых числа a и b . Для этих чисел выполняются условия $-10^9 \leq a, b \leq 10^9$. Выход: единственное целое число — результат сложения $a + b^2$.

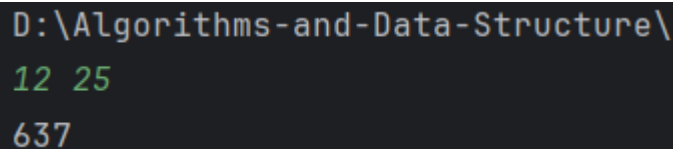
Код:

```
_min = 10 ** (-9)
_max = 10 ** 9
a, b = map(int, input().split())
if (_min <= a <= _max) and (_min <= b <= _max):
    print(a + b ** 2)
else:
    print('Введены некорректные данные')
```

Текстовое объяснение решения:

- 1) Заданы левая и правая границы интервала допустимых значений.
- 2) Считываются два числа, введенные пользователем.
- 3) Числа проверяются на соответствие интервалу, при выполненном условии выводится ответ на задачу.

Примеры работы кода:



```
D:\Algorithms-and-Data-Structure\
12 25
637
```

Вывод по задаче:

В решении задачи так же использованы переменные, операторы сравнение и if-else конструкции, но в отличие от первой задачи здесь одно из чисел возводится в степень. Код выполняет поставленную задачу.

Задание №1.3. $a+b$ с использованием файлов

Текст задачи:

3. Выполните задачу $a + b$ с использованием файлов.

- Имя входного файла: input.txt
- Имя выходного файла: output.txt
- Формат входного файла. Входной файл состоит из одной строки, которая содержит два целых числа a и b . Для этих чисел выполняются условия $-10^9 \leq a, b \leq 10^9$.
- Формат выходного файла. Выходной файл единственное целое число — результат сложения $a + b$.

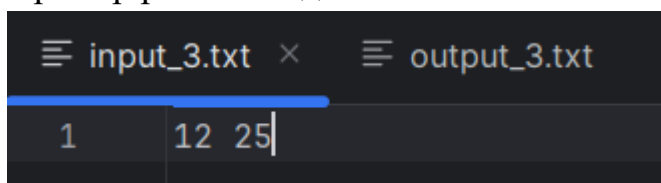
Код:

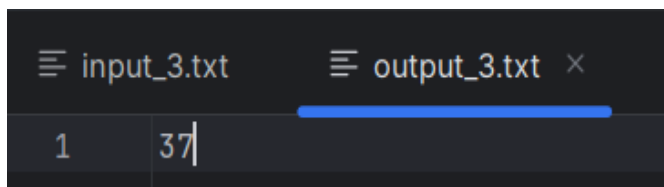
```
_min = 10 ** (-9)
_max = 10 ** 9
f = open('input_3.txt')
a, b = map(int, f.readline().split())
f.close()
if (_min <= a <= _max) and (_min <= b <= _max):
    f_output = open('output_3.txt', 'w')
    f_output.write(str(a + b))
    f_output.close()
else:
    f_output = open('output_3.txt', 'w')
    f_output.write('Введены некорректные данные')
    f_output.close()
```

Текстовое объяснение решения:

- 1) Заданы левая и правая границы интервала допустимых значений.
- 2) Считываются два числа из файла «input_3.txt».
- 3) Числа проверяются на соответствие интервалу, при выполненном условии ответ на задачу записывается в файл «output_3.txt».

Пример работы кода:





Вывод по задаче:

В решении задачи использованы переменные, операторы сравнение и if-else конструкции, а также работа с файлами. Задача подобна первой, за исключением ввода-вывода в файлы. Код выполняет поставленную задачу.

Задание №1.4. $a+b^2$ с использованием файлов

Текст задачи:

4. Выполните задачу $a+b^2$ с использованием файлов аналогично предыдущему пункту.

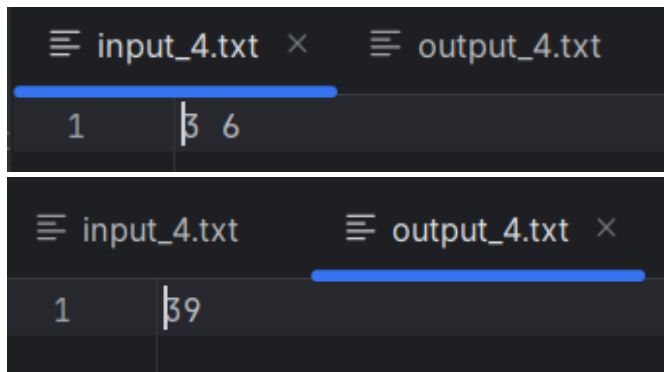
Код:

```
_min = 10 ** (-9)
_max = 10 ** 9
f = open('input_4.txt')
a, b = map(int, f.readline().split())
f.close()
if (_min <= a <= _max) and (_min <= b <= _max):
    f_output = open('output_4.txt', 'w')
    f_output.write(str(a + b ** 2))
    f_output.close()
else:
    f_output = open('output_4.txt', 'w')
    f_output.write('Введены некорректные данные')
    f_output.close()
```

Текстовое объяснение решения:

- 1) Заданы левая и правая границы интервала допустимых значений.
- 2) Считываются два числа из файла «input_4.txt».
- 3) Числа проверяются на соответствие интервалу, при выполненном условии ответ на задачу записывается в файл «output_4.txt».

Пример работы кода:



Вывод по задаче:

В решении задачи использованы переменные, операторы сравнение и if-else конструкции, а также работа с файлами. Задача подобна первой, за исключением ввода-вывода в файлы. Код выполняет поставленную задачу.

Задание №2. Число Фибоначчи.

Текст задачи:

Задание 2. Число Фибоначчи

Определение последовательности Фибоначчи:

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_i &= F_{i-1} + F_{i-2} \text{ для } i \geq 2. \end{aligned} \tag{1}$$

Таким образом, каждое число Фибоначчи представляет собой сумму двух предыдущих, что дает последовательность

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

Ваша цель – разработать эффективный алгоритм для подсчета чисел Фибоначчи. Вам предлагается начальный код на Python, который содержит наивный рекурсивный алгоритм:

```
def calc_fib(n):
    if (n <= 1):
        return n

    return calc_fib(n - 1) + calc_fib(n - 2)

n = int(input())
print(calc_fib(n))
```

- Имя входного файла: input.txt
- Имя выходного файла: output.txt
- Формат входного файла. Целое число n . $0 \leq n \leq 45$.
- Формат выходного файла. Число F_n .

Активировать
чтобы
"Править"

Код:

```
import time
```

```
t_start = time.perf_counter()
```

```
f = open('input.txt')
```

```
n = int(f.readline())
```

```
f.close()
```

```
_min = 0
```

```
_max = 45
```

```
if _min <= n <= _max:
```

```
    f_0, f_1 = 0, 1
```

```
    for i in range(n - 1):
```



```

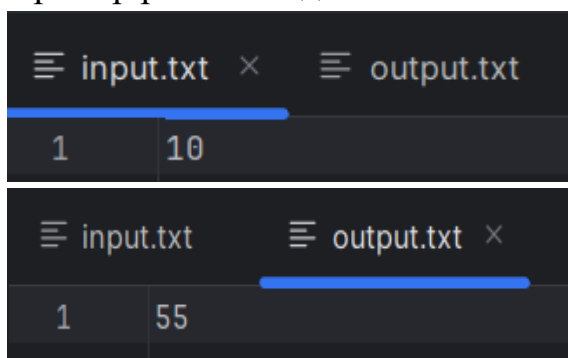
    f_0, f_1 = f_1, f_0 + f_1
    f_output = open('output.txt', 'w')
    f_output.write(str(f_1))
    f_output.close()
    print("Время работы: %s секунд " % (time.perf_counter() - t_start))
else:
    f_output = open('output.txt', 'w')
    f_output.write('Введены некорректные данные')
    f_output.close()

```

Текстовое объяснение решения:

- 1) Импорт модуля time для отслеживания времени работы программы.
- 2) Объявление переменной счетчика времени.
- 3) Считывается число из файла «input.txt».
- 4) Число проверяется на соответствие интервалу, при выполненном условии ответ на задачу записывается в файл «output.txt». Циклом for считается нужное число Фибоначчи.
- 5) Последней строчкой выводится время работы алгоритма.

Пример работы кода:



	Время выполнения
Нижняя граница диапазона значений входных данных из текста задачи	Время работы: 0.0015264999819919467 секунд
Пример из задачи	Время работы: 0.0013092000153847039 секунд

Вывод по задаче:

В решении задачи использованы переменные,

Верхняя граница диапазона значений входных данных из текста задачи	Время работы: 0.0012135999859310687 секунд	операторы сравнение, if-else конструкции, цикл for, а также работа с файлами. При решении задачи
--------------------------------------------------------------------	--------------------------------------------	--------------------------------------------------------------------------------------------------

понадобилось разобрать алгоритм Фибоначчи, а так же посчитать время работы алгоритма. Код выполняет поставленную задачу.

Задание №3. Еще про числа Фибоначчи

Текст задачи:

Задание 3. Еще про числа Фибоначчи

Определение последней цифры большого числа Фибоначчи. Числа Фибоначчи растут экспоненциально. Например,

$$F_{200} = 280571172992510140037611932413038677189525$$

Хранить такие суммы в массиве, и при этом подсчитывать сумму, будет достаточно долго. Найти последнюю цифру любого числа достаточно просто: $F \bmod 10$.

- Имя входного файла: input.txt
- Имя выходного файла: output.txt
- Формат входного файла. Целое число n . $0 \leq n \leq 10^7$.
- Формат выходного файла. Одна последняя цифра числа F_n .

Код:

```
import time

t_start = time.perf_counter()
f = open('input.txt')
n = int(f.readline())
f.close()
_min = 0
_max = 10 ** 7
if _min <= n <= _max:
    f_0, f_1 = 0, 1
    for i in range(n - 1):
```

```

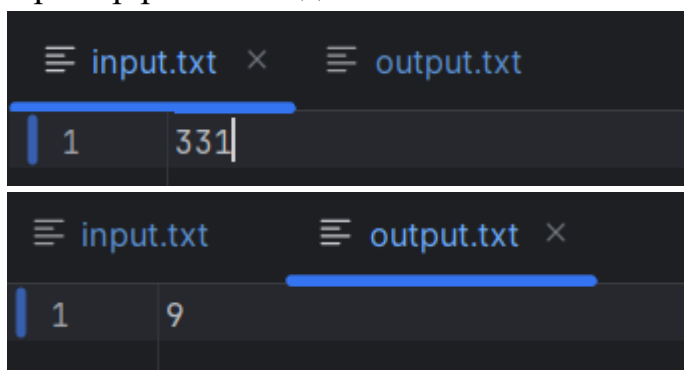
    f_0, f_1 = f_1 % 10, (f_0 + f_1) % 10
    f_output = open('output.txt', 'w')
    f_output.write(str(f_1))
    f_output.close()
    print("Время работы: %s секунд " % (time.perf_counter() - t_start))
else:
    f_output = open('output.txt', 'w')
    f_output.write('Введены некорректные данные')
    f_output.close()

```

Текстовое объяснение решения:

- 1) Импорт модуля time для отслеживания времени работы программы.
- 2) Объявление переменной счетчика времени.
- 3) Считывается число из файла «input.txt».
- 4) Число проверяется на соответствие интервалу, при выполненном условии ответ на задачу записывается в файл «output.txt». Циклом for считается остатки от деления на 10 чисел Фибонначи.
- 5) Последней строчкой выводится время работы алгоритма.

Пример работы кода:



	Время выполнения
Нижняя граница диапазона значений входных данных из текста задачи	Время работы: 0.003492899995762855 секунд
Пример из задачи	Время работы: 0.0016191999893635511 секунд

Пример из задачи	Время работы: 0.05293469998287037 секунд
Верхняя граница диапазона значений входных данных из текста задачи	Время работы: 1.5120806999620982 секунд

Вывод по задаче:

В решении задачи использованы переменные, операторы сравнение, if-else конструкции, цикл for, а также работа с файлами. При решении задачи понадобилось разобрать алгоритм Фибоначчи и брать только остатки от деления на 10, а так же посчитать время работы алгоритма. Код выполняет поставленную задачу.

Вывод:

В данной лабораторной работе я практиковалась в работе с файлами, с логическими конструкциями, с модулем time, для оценки скорости работы алгоритмов.