

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

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

Выполнила:
Пожидаева Е.Р.

Санкт-Петербург

2024 г.

Задание № 1. Ввод-вывод.

Текст задачи

Вам необходимо выполнить 4 следующих 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 $a + b$.
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 $a + b^2$.
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 $a + b$.

Примеры.

input.txt	12 25	130 61
output.txt	37	191

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

Листинг кода.

```
a, b = map(int, input().split())
print(a+b)
```

```
a, b = map(int, input().split())
print(a + b**2)
```

```
f = open('input.txt', 'r')
s = f.read()
a, b = map(int, s.split())
summa = str(a + b)

f = open('output.txt', 'w')
f.write(summa)
f.close()

print(end)
```

```

f = open('input.txt', 'r')
s = f.read()
a, b = map(int, s.split())
summa = str(a + b**2)

f = open('output.txt', 'w')
f.write(summa)
f.close()

print(end)

```

Результат работы кода на примерах из текста задачи:

```

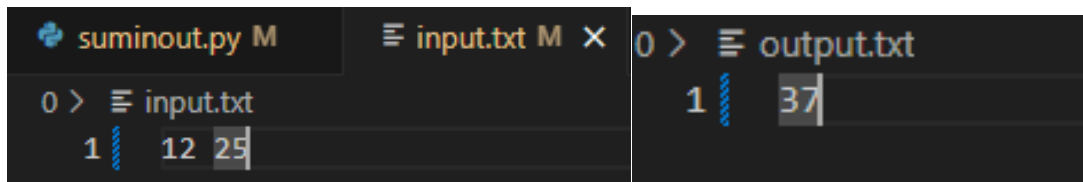
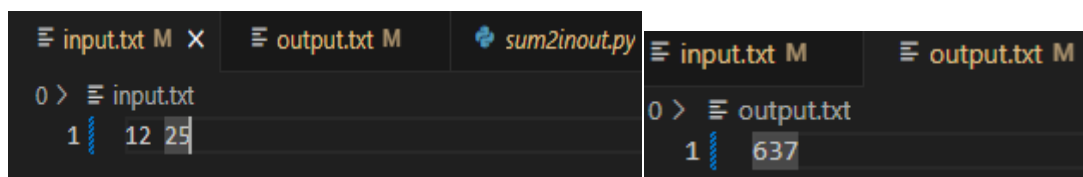
PS C:\Users\user\Desktop\Academ_zadol\7\4_zad> & C:/Users/user/AppData/Local/Microso
ft/WindowsApps/python3.11.exe c:/Users/user/Desktop/Academ_zadol/0/zadaniye1.py
12 25
37

```

```

PS C:\Users\user\Desktop\Academ_zadol\7\4_zad> & C:/Users/user/AppData/Local/Microso
ft/WindowsApps/python3.11.exe c:/Users/user/Desktop/Academ_zadol/0/zadan1.2.py
12 25
637

```

Вывод по задаче: научилась писать программу, которая способна взаимодействовать с пользователем

Задание № 2. Число Фибоначчи.

Текст задания.

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 .
- Пример.

input.txt	10
output.txt	55

Листинг

```
import time
import psutil
mem = psutil.Process().memory_info().rss
start = time.time()

f = open('input.txt')
n = int(f.readline())
a = [0, 1]
for i in range(n):
    a.append(a[i] + a[i + 1])
```

```
f = open('output1.txt', 'w')
f.write(str(a[n]))
f.close()
```

```
end = time.time() - start
print('{}'.format(mem))
print(end)
```

Результат работы кода на примерах из текста задачи:

The screenshot shows a code editor with two files open. The first file, `input.txt`, contains a list of numbers: `1` and `10`. The second file, `fibinput.py`, contains a list of numbers: `1` and `55`. Both files are shown with their respective tabs and file names.

	Время выполнения (сек)	Затраты памяти (Мб)
Пример из задачи	0.004347085952758789	22474752

Вывод по задаче: разработала эффективный алгоритм для подсчета чисел Фибоначчи и протестировала время выполнения алгоритма

Задание № 3. Еще про числа Фибоначчи.

Текст задачи

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 .
- Пример 1.

input.txt	331
output.txt	9

$$F_{331} = 668996615388005031531000081241745415306766517246774551964595292186469.$$

- Пример 2.

input.txt	327305
output.txt	5

Это число не влезет в страницу, но оканчивается действительно на 5.

- Ограничение по времени: 5сек.
- Ограничение по памяти: 512 мб.

Листинг

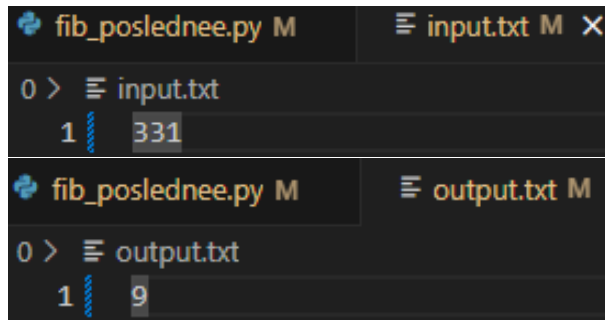
```
import time
import psutil
mem = psutil.Process().memory_info().rss
start = time.time()

f = open('input.txt')
n = int(f.readline())
a = [0, 1]
for i in range(n - 1):
    a.append((a[i] + a[i + 1]) % 10)
f = open('output.txt', 'w')
f.write(str(a[n]))
f.close()

end = time.time() - start
print('{}'.format(mem))
```

```
print(end)
```

Результат работы кода на примерах из текста задачи:



```
fib_poslednee.py M  input.txt M X
0 > input.txt
1 331
fib_poslednee.py M  output.txt M
0 > output.txt
1 9
```

	Время выполнения (сек)	Затраты памяти (Мб)
Пример из задачи	0.004905939102172851 6	22159360

Вывод по задаче: научилась определять последнюю цифру большого числа Фибоначчи и протестировала время выполнения алгоритма.