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:

Кретов Иван

К3139

Проверил:

Афанасьев А.В.

Санкт-Петербург

2024 г.

# Содержание отчета

[**Содержание отчета**](#_heading=h.gjdgxs) **2**

[**Задачи по варианту**](#_heading=h.30j0zll) **3**

[Задача №1.](#_heading=h.1fob9te) Ввод-вывод 3

Задача №2. Число Фибоначчи 7

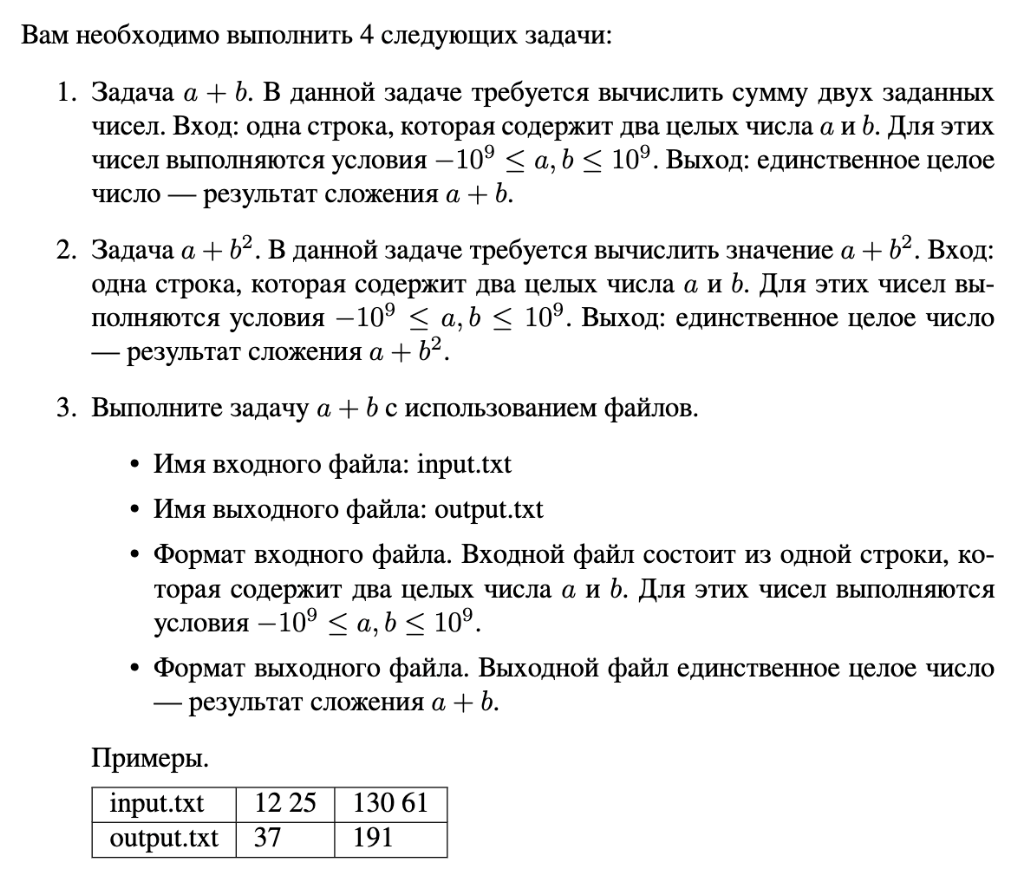
Задача №3. Ещё про числа Фибоначчи 10

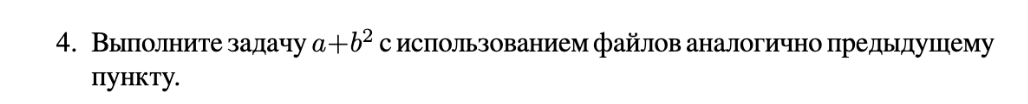
Задача №4. Тестирование ваших алгоритмов 13

[**Вывод**](#_heading=h.tyjcwt) **14**

# Задачи по варианту

## Задача №1. Ввод-вывод





**1.1**

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

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

print(a+b)

else:

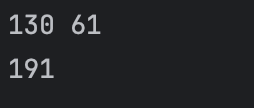
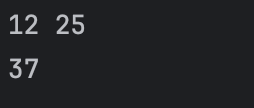
print("ERROR")

1. После ввода двух чисел превращаю строку в лист с помощью функции split, и функцией map превращаю каждый элемент в int

2. Далее проверяю подходят ли переменные под ограничение задания и вывожу сумму переменных

3. В противном случае вывожу “ERROR”

Результат работы кода:



**1.2**

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

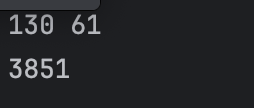
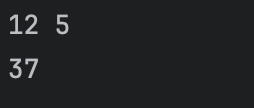
**if -10\*\*9 <= a <= 10\*\*9 and -10\*\*9 <= b <= 10\*\*9:**

**print(a+b\*b)**

**else:**

**print("ERROR")**

1. Аналогично с 1.1



**1.3**

f = open('input.txt')

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

f.close()

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

f = open('output.txt', 'w')

s = str(a+b)

f.write(s)

f.close()

else:

print("ERROR")

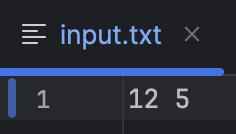
1. Открываю файл input.txt

2. Читаю строку из файла и превращаю ее в лист с помощью функции split, и функцией map превращаю каждый элемент в int

3. Проверяю подходят ли числа под ограничения, открываю файл output.txt и записываю в него результат сложения двух переменных

4. В противном случае вывожу “ERROR”

Входные данные:



Результат:



**1.4**

f = open('input.txt')

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

f.close()

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

f = open('output.txt', 'w')

s = str(a+b\*b)

f.write(s)

f.close()

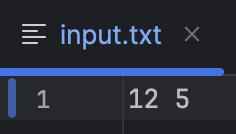
else:

print("ERROR")

1.1 Аналогично с 1.3



Входные данные:



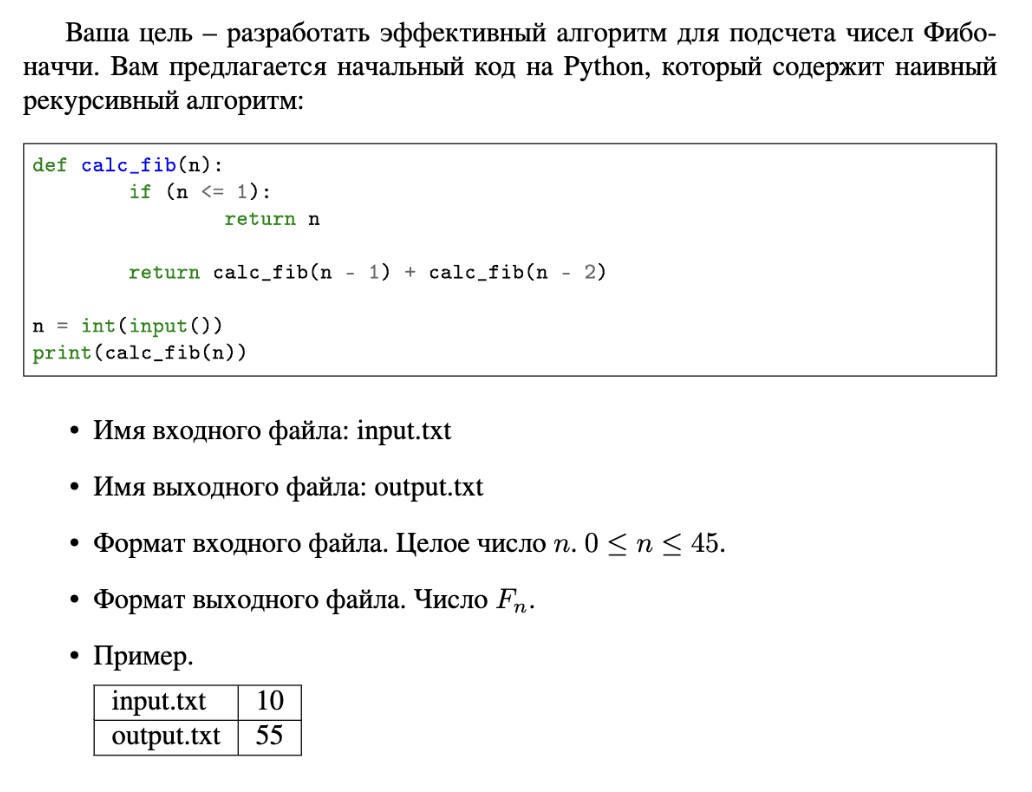
Результат:

## 

## 

## 

## Задача №2. Число Фибоначчи



import time

t\_start = time.perf\_counter()

f = open('input.txt')

k = int(f.readline())

f.close()

if 0 < k <= 45:

a, b = 0, 1

for i in range(k-1):

a, b = b, a+b

f = open('output.txt', 'w')

f.write(str(b))

f.close()

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

elif k == 0:

f = open('output.txt', 'w')

f.write("0")

f.close()

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

else:

print("ERROR")

1. Задаю в переменную время начала выполнения алгоритма

2. Открываю файл input.txt и читаю из него число

3. Проверяю подходит ли оно под условие

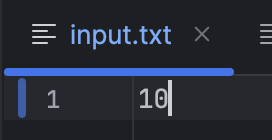
4. С помощью алгоритма высчитываю нужно число Фибоначчи

5. Открываю файл output.txt и вписываю туда ответ и вывожу время за которое была выполнен код

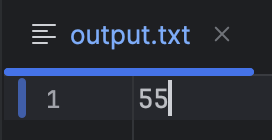
6. Если на вход даётся число “0” то записываю в файл ответ: “0” и вывожу время за которое была выполнен код

7. Если введено число не подходящее под условие то вывожу “ERROR”

Входные данные:



Результат:





|  | Время выполнения |
| --- | --- |
| Нижняя граница диапазона значений входных данных из текста задачи | 0.0001944169998751022 секунд |
| Пример из задачи | 0.00024745799601078033 секунд |
| Верхняя граница диапазона значений входных данных из текста задачи | 0.00032483399991178885 секунд |

## Задача №3. Ещё про числа Фибоначчи

## 

import time

t\_start = time.perf\_counter()

f = open('input.txt')

k = int(f.readline())

f.close()

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

a, b = 0, 1

for i in range(k-1):

a, b = b, (a+b)%10

f = open('output.txt', 'w')

f.write(str(b))

f.close()

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

elif k == 0:

f = open('output.txt', 'w')

f.write("0")

f.close()

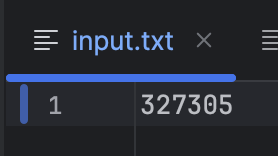
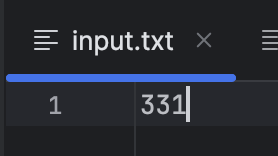
print("Время работы: %s секунд " % (time.perf\_counter() - t\_start))

else:

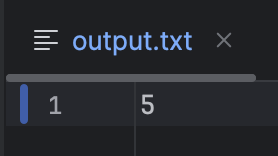
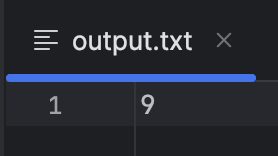
print("ERROR")

1. Аналогично 2 задаче, однако нам не обязательно находить целое число Фибоначчи, достаточно найти последнюю цифру, из-за чего немного меняется алгоритм

Входные данные:

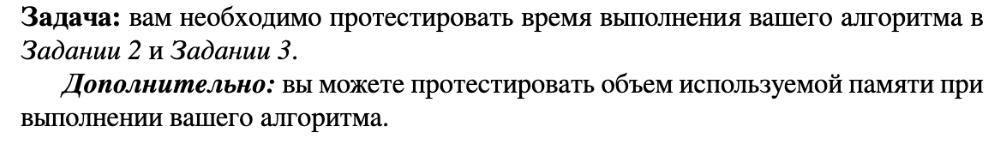


Результат:



|  | Время выполнения |
| --- | --- |
| Нижняя граница диапазона значений входных данных из текста задачи | 0.0003292499968665652 секунд |
| Пример из задачи | 0.0002703749996726401 секунд |
| Пример из задачи | 0.02684033399418695 секунд |
| Верхняя граница диапазона значений входных данных из текста задачи | 0.8528328750035143 секунд |

## Задача №4. Тестирование ваших алгоритмов



Задача №2:

import time

t\_start = time.perf\_counter()

f = open('input.txt')

k = int(f.readline())

f.close()

if 0 < k <= 45:

a, b = 0, 1

for i in range(k-1):

a, b = b, a+b

f = open('output.txt', 'w')

f.write(str(b))

f.close()

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

elif k == 0:

f = open('output.txt', 'w')

f.write("0")

f.close()

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

else:

print("ERROR")

Задача №3:

import time

t\_start = time.perf\_counter()

f = open('input.txt')

k = int(f.readline())

f.close()

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

a, b = 0, 1

for i in range(k-1):

a, b = b, (a+b)%10

f = open('output.txt', 'w')

f.write(str(b))

f.close()

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

elif k == 0:

f = open('output.txt', 'w')

f.write("0")

f.close()

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

else:

print("ERROR")

1. Работа каждого кода была описана в соответствующем для задачи заголовке

Время выполнения алгоритма в Задании 2:



Время выполнения алгоритма в Задании 3:



# 

# Вывод (по всей лабораторной):

1. В ходе лабораторной работы вспомнил как читать и записывать данные из файла
2. Был написан алгоритм нахождения n-ого числа Фибоначчи
3. Был изучен способ контроля времени работы алгоритма