

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

Выполнил:  
Авдиенко Данила Андреевич  
Группа К3140

Проверил:  
Афанасьев А. В.

Санкт-Петербург  
2024 г.

# Содержание отчета

## Оглавление

<i>Содержание отчета</i> .....	2
<i>Задачи по варианту</i> .....	3
Задание № 1.1. $a+b$ .....	3
Задание №1.2. $a+b^2$ .....	4
Задание №1.3. $a+b$ с использованием файлов .....	5
Задание №1.4. $a+b^2$ с использованием файлов.....	6
Задание №2. Число Фибоначчи. ....	7
Задание №3. Еще про числа Фибоначчи .....	9
Вывод: .....	11

## 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:

```
b1 = 10 ** (-9)
br = 10 ** 9
# Задача 1
a1, b1 = map(int, input().split())
flag = False
while flag != True:
    if (b1 <= a1 <= br) and (b1 <= b1 <= br):
        print(a1 + b1)
        flag = True
    else:
        print("Введенные данные не соответствуют условию. Введите новые")
        a1, b1 = map(int, input().split())
```

Текстовое объяснение решения.

- 1) Заданы левая и правая границы интервала допустимых значений ( $b1$  и  $br$ ).
- 2) Считываются два числа, введенных пользователем.
- 3) Числа проверяются на соответствие интервалу, если условие выполнено, то выводится их сумма, если условие не выполнено, то выводится соответствующее сообщение.

Примеры работы кода:

```
/Users/danilaavdienko/PycharmProjects/pythonProject/venv/bin/python
12 25
37
Process finished with exit code 0

/Users/danilaavdienko/PycharmProjects/pythonProject/venv/bin/python
130 61
191
Process finished with exit code 0
```

Вывод по задаче:

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

## Задание №1.2. $a+b^2$

Текст задачи.

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:

```
b1 = 10 ** (-9)
br = 10 ** 9
# Задача 2
a2, b2 = map(int, input().split())
flag2 = False
while flag2 != True:
    if (b1 <= a2 <= br) and (b1 <= b2 <= br):
        print(a2 + (b2 ** 2))
        flag2 = True
    else:
        print("Введенные данные не соответствуют условию. Введите новые")
        a2, b2 = map(int, input().split())
```

Текстовое объяснение решения.

- 1) Ранее заданы левая и правая границы интервала допустимых значений (b1 и br).
- 2) Считываются два числа, введенных пользователем.
- 3) Числа проверяются на соответствие интервалу, если условие выполнено, то выводится сумма первого числа и квадрата второго, если условие не выполнено, то выводится соответствующее сообщение.

Примеры работы кода:

```
/Users/danilaavdienko/PycharmProjects/pythonProject/venv/bin/python
130 61
3851

Process finished with exit code 0

/Users/danilaavdienko/PycharmProjects/pythonProject/venv/bin/python
12 25
637

Process finished with exit code 0
```

Вывод по задаче:

В решении задачи так же использованы переменные, if-else конструкции и операторы сравнения, в отличие от первой задачи, здесь потребовалось возвести одно из чисел во вторую степень. Код выполняет поставленную задачу.

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

Текст задачи:

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:

```
# Задача 3
f = open("input.txt")
line = f.readline()
a, b = map(int, line.split())
if (bl <= a <= br) and (bl <= b <= br):
    answ = str(a + b)
    fo = open("output.txt", "w")
    fo.write((answ))
else:
    print("Введенные данные не соответствуют условию")
```

Текстовое объяснение решения.

- 1) Ранее заданы левая и правая границы интервала допустимых значений (bl и br).
- 2) Открывается файл input.txt с входными данными.
- 3) Считывается первая строка входного файла, из неё берутся числа  $a$  и  $b$ .
- 4) Числа проверяются на соответствие интервалу, если условие выполнено, то в переменную answ вводится значение суммы чисел в строковом виде, открывается файл output.txt и в него записывается переменная answ. Если условие не выполнено, то выводится соответствующее сообщение.

Пример работы кода:

input.txt ×			:	output.txt ×		
1	130	61	✓	1	191	

Вывод по задаче:

Эта задача подобна первой, но требует умения работы с файлами (открытия, считывания и записи данных). 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:

```
bl = 10 ** (-9)
br = 10 ** 9
# Задача 4
fi = open("input.txt")
line = fi.readline()
a, b = map(int, line.split())
if (bl <= a <= br) and (bl <= b <= br):
    answ = str(a + b ** 2)
    fo = open("output.txt", "w")
    fo.write(answ)
else:
    print("Введенные данные не соответствуют условию")
```

Текстовое объяснение решения.

- 1) Ранее заданы левая и правая границы интервала допустимых значений (bl и br).
- 2) Открывается файл input.txt с входными данными.
- 3) Считывается первая строка входного файла, из неё берутся числа a и b.
- 4) Числа проверяются на соответствие интервалу, если условие выполнено, то в переменную answ вводится значение суммы первого числа и квадрата второго в строковом виде, открывается файл output.txt и в него записывается переменная answ. Если условие не выполнено, то выводится соответствующее сообщение.

Пример работы кода:

input.txt ×			:	output.txt ×		
1	130	61	✓	1	3851	

Вывод по задаче:

Эта задача подобна второй, но требует умения работы с файлами (открытия, считывания и записи данных). Код выполняет поставленную задачу.

## Задание №2. Число Фибоначчи.

Текст задачи:

### Задание 2. Число Фибоначчи

Определение последовательности Фибоначчи:

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

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

Код:

```
import time

t_start = time.perf_counter()
fi = open("input.txt")
l = fi.readline()
a = int(l)
sum = 0
if not (0 <= a <= 45):
    print("Число не удовлетворяет границам")
    exit()

def calc_fibonacci(a):
    l = [0, 1]
    i = 2
    while i <= a:
        l.append(l[i - 1] + l[i - 2])
        i += 1
    return l[a]

answ = str(calc_fibonacci(a))
fo = open("output.txt", "w")
fo.write(answ)

print("Время работы алгоритма:", (time.perf_counter() - t_start))
```

Текстовое объяснение решения.

- 1) Импорт модуля `time` с последующим объявлением переменной-счетчика для отслеживания времени работы алгоритма.
- 2) Открывается файл `input.txt` с входными данными.
- 3) Считывается первая строка входного файла, из неё берется число `a` и проверяется на соответствие интервалу, если условие не выполняется, то код завершается преждевременно.
- 4) Далее описана функция `calc_fibonacci`, которая итеративно подсчитывает конкретное число Фибоначчи.
- 5) После подсчета в переменную `answ` вводится число Фибоначчи, открывается файл `output.txt` и в него записывается переменная `answ`.
- 6) Крайней командой выводится время выполнения алгоритма.

Пример работы кода:

≡ input.txt	▼	:	≡ output.txt	×
1	10	✓	1	55

	Время выполнения
Нижняя граница диапазона значений входных данных из текста задачи	Время работы алгоритма: 0.00013929099999999972
Пример из задачи	Время работы алгоритма: 0.00012449999999999996
Верхняя граница диапазона значений входных данных из текста задачи	Время работы алгоритма: 0.00010545800000000098

Вывод по задаче:

При решении задачи потребовалось применить полученные ранее знания по работе с файлами, разобрать алгоритм для подсчета числа Фибоначчи и 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  $F_n$ .
- 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.

### Код:

```
import time

t_start = time.perf_counter()
fi = open("input.txt")
l = fi.readline()
a = int(l)
if not (0 <= a <= 10 ** 7):
    print("Число не удовлетворяет границам")
    exit()

def lastD_fib(n):
    if n == 0:
        return 0
    a = 0
    b = 1
    for i in range(n - 1):
        c = (a + b) % 10
        a, b = b, c
    return b

answ = str(lastD_fib(a))
fo = open("output.txt", "w")
fo.write(answ)

print("Время работы алгоритма:", (time.perf_counter() - t_start))
```

Текстовое объяснение решения.

- 1) Импорт модуля `time` с последующим объявлением переменной-счетчика для отслеживания времени работы алгоритма.
- 2) Открывается файл `input.txt` с входными данными.
- 3) Считывается первая строка входного файла, из неё берется число `a` и проверяется на соответствие интервалу, если условие не выполняется, то код завершается преждевременно.
- 4) Далее описана функция `lastD_fib`, которая вычисляет только последние цифры чисел Фибоначчи, это позволяет обрабатывать более крупные числа.
- 5) После подсчета в переменную `answ` вводится последняя цифра числа Фибоначчи, открывается файл `output.txt` и в него записывается переменная `answ`.
- 6) Крайней командой выводится время выполнения алгоритма.

	Время выполнения
Нижняя граница диапазона значений входных данных из текста задачи	Время работы алгоритма: 0.00013220799999999998
Пример из задачи	Время работы алгоритма: 0.000165708000000000052
Пример из задачи	Время работы алгоритма: 0.013591124999999999
Верхняя граница диапазона значений входных данных из текста задачи	Время работы алгоритма: 0.364791083

Вывод по задаче:

Задача потребовала значительной модификации функции, написанной для задания №3. В этом коде так же использовались логические конструкции, работа с файлами и создание функции. Код выполняет поставленную задачу.

**Вывод:**

В данной лабораторной работе я попрактиковался в использовании логических конструкций, вспомнил работу с файлами, научился работать с модулем `time` для наглядной оценки скорости работы алгоритма, познакомился с рекурсивным и итеративным алгоритмами для вычисления чисел Фибоначчи.