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

[Содержание отчета 2](#_Toc177731738)

[Задачи по варианту 3](#_Toc177731739)

[Задание № 1.1. a+b 3](#_Toc177731740)

[Задание №1.2. a+b2 4](#_Toc177731741)

[Задание №1.3. a+bс использованием файлов 5](#_Toc177731742)

[Задание №1.4. a+b2 с использованием файлов 6](#_Toc177731743)

[Задание №2. Число Фибоначчи. 7](#_Toc177731744)

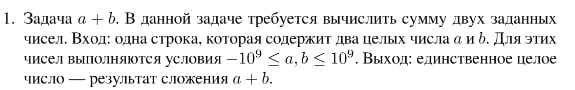
[Задание №3. Еще про числа Фибоначчи 9](#_Toc177731745)

[Вывод: 11](#_Toc177731746)

# Задачи по варианту

## Задание № 1.1. a+b

Текст задачи.



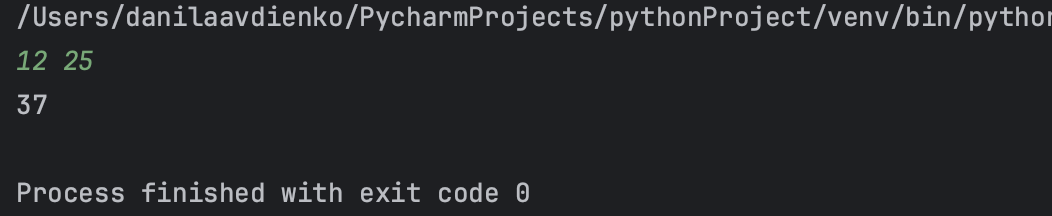
Код:

bl = 10 \*\* (-9)  
br = 10 \*\* 9  
# Задача 1  
a1, b1 = map(int, input().split())  
flag = False  
while flag != True:  
 if (bl <= a1 <= br) and (bl <= 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 (bl 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:

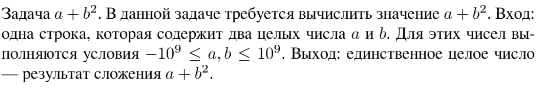


Вывод по задаче:

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

## Задание №1.2. a+b2

Текст задачи.



Код:

bl = 10 \*\* (-9)  
br = 10 \*\* 9

# Задача 2  
a2, b2 = map(int, input().split())  
flag2 = False  
while flag2 != True:  
 if (bl <= a2 <= br) and (bl <= 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 (bl 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:



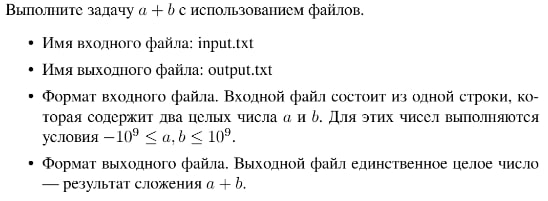


Вывод по задаче:

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

## Задание №1.3. a+bс использованием файлов

Текст задачи:



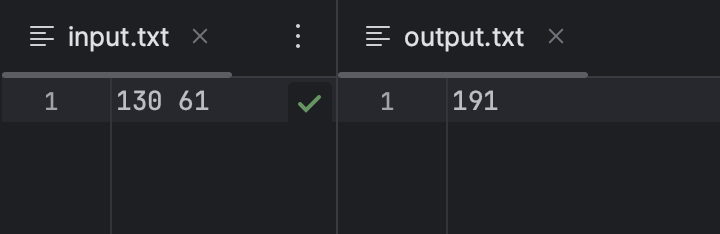
Код:

# 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:



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+b2 с использованием файлов

Текст задачи.



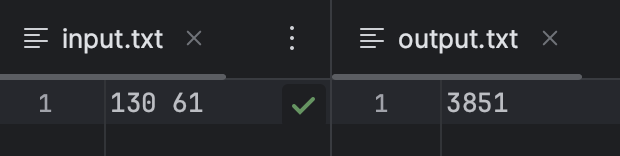
Код:

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:

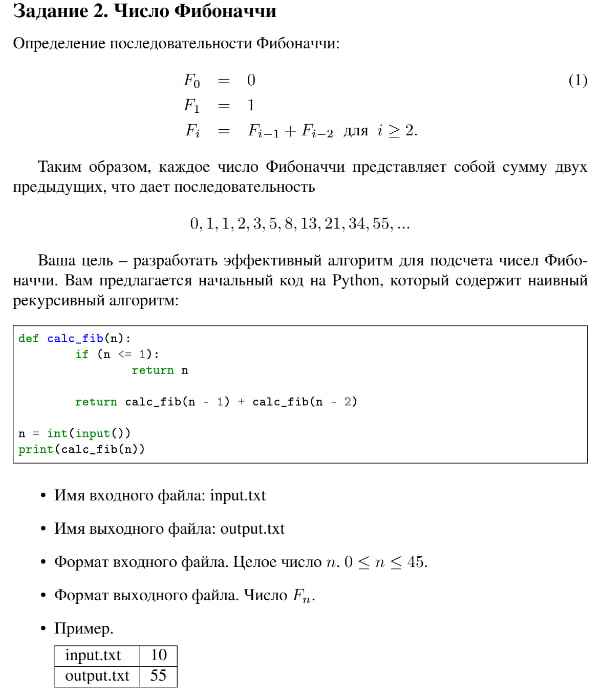


Вывод по задаче:

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:



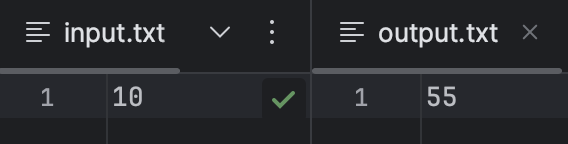
Код:

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:



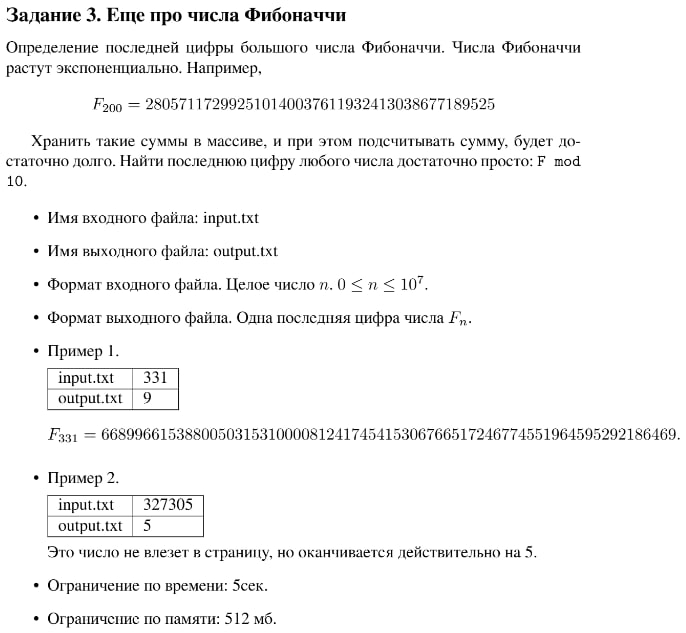
|  |  |
| --- | --- |
|  | Время выполнения |
| Нижняя граница диапазона значений входных данных из текста задачи | Время работы алгоритма: 0.00013929099999999972 |
| Пример из задачи | Время работы алгоритма: 0.0001244999999999996 |
| Верхняя граница диапазона значений входных данных из текста задачи | 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:



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 цифры 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: 0.00016570800000000052 |
| 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.