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. Работа с файлами, тестирование

Выполнил:

Аксянова А.Р

К3140

Проверил:

Афанасьев А.В

Санкт-Петербург

2024 г.

# Содержание отчета

[Содержание отчета 2](#_Toc177329332)

[Задачи 3](#_Toc177329333)

[Задача №1. Ввод-вывод 3](#_Toc177329334)

[Задача 1.1 3](#_Toc177329335)

[Задача 1.2 3](#_Toc177329336)

[Задача 1.3 4](#_Toc177329337)

[Задача 1.4 6](#_Toc177329338)

[Задача №2. Число Фибоначчи 6](#_Toc177329339)

[Задача №3. Еще про числа Фибоначчи 8](#_Toc177329340)

[Задача №4. Тестирование ваших алгоритмов 11](#_Toc177329341)

[Задача 4.1( тест задачи 2) 11](#_Toc177329342)

[Задача 4.2 ( тест задачи 3) 13](#_Toc177329343)

[Вывод 15](#_Toc177329344)

# Задачи

## Задача №1. Ввод-вывод

### Задача 1.1

1. Текст задачи

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 −109 ≤ *a,b* ≤109. Выход: единственное целое число — результат сложения *a* + *b*.

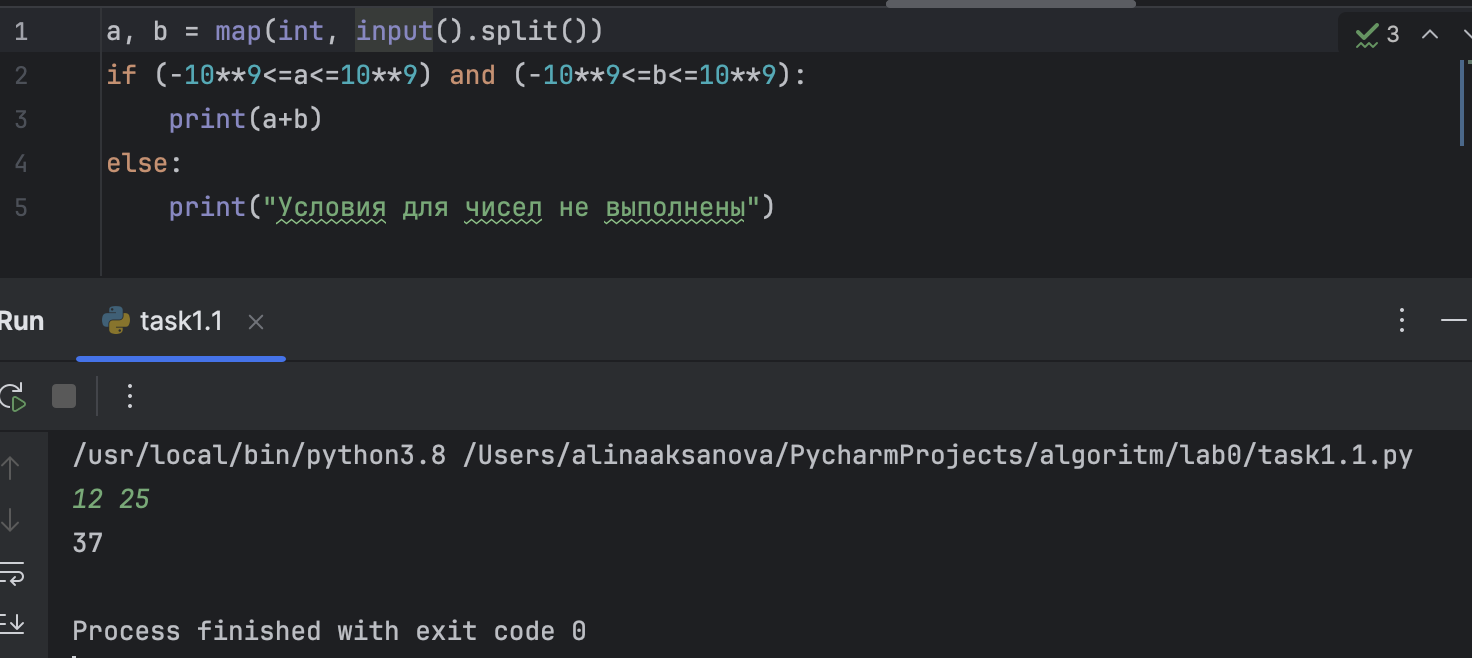
1. Листинг кода

a, b = map(int, input().split())  
if (-10\*\*9<=a<=10\*\*9) and (-10\*\*9<=b<=10\*\*9):  
 print(a+b)  
else:  
 print("Условия для чисел не выполнены")

1. Текстовое объяснение решения.

С помощью функции map я считываю числа a и b, данные для входа на одной строке. Далее проверяю, удовлетворяют ли они условию: если да, то вывожу результат сложения этих чисел, в противном случае печатаю “Условия для чисел не выполнены”.

1. Результат работы кода на примерах из текста задачи:



1. Вывод по задаче:

Я вспомнила как писать базовый код на Python, как считывать данные и проверять наличие выполнения условия.

### Задача 1.2

1. 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 −109 ≤ *a,b* ≤ 109. Выход: единственное целое число — результат сложения *a* + *b*2.

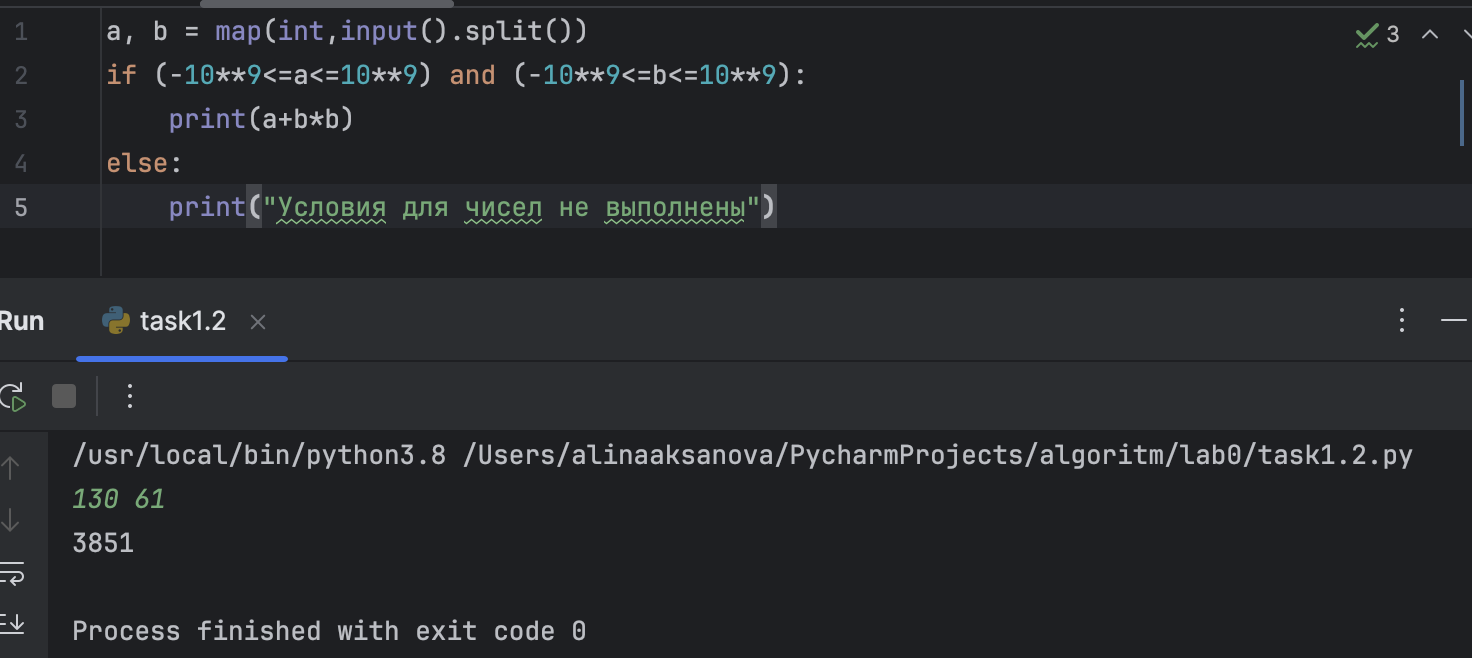
1. Листинг кода

a, b = map(int, input().split())  
if (-10\*\*9<=a<=10\*\*9) and (-10\*\*9<=b<=10\*\*9):  
 print(a+b\*b)  
else:  
 print("Условия для чисел не выполнены")

1. Текстовое объяснение решения.

С помощью функции map я считываю числа a и b, данные для входа на одной строке. Далее проверяю, удовлетворяют ли они условию: если да, то вывожу результат сложения числа a с квадратом числа b, в противном случае печатаю “Условия для чисел не выполнены”.

1. Результат работы кода на примерах из текста задачи:



1. Вывод по задаче:

Я вспомнила как писать базовый код на Python, как считывать данные и проверять наличие выполнения условия.

### Задача 1.3

1. 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 −109 ≤ *a,b* ≤109.
* Формат выходного файла. Выходной файл единственное целое число — результат сложения *a* + *b*.

Примеры.

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

1. Листинг кода

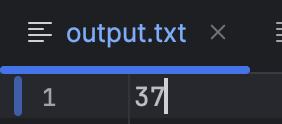
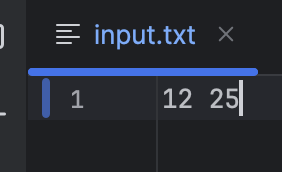
f = open("input.txt")  
a,b = map(int, f.readline().split())  
if (-10\*\*9<=a<=10\*\*9) and (-10\*\*9<=b<=10\*\*9):  
 summ = str(a+b)  
 k = open("output.txt", "w")  
 k.write(summ)  
 print(a+b)  
else:  
 print("Условия для чисел не выполнены")

1. Текстовое объяснение решения.

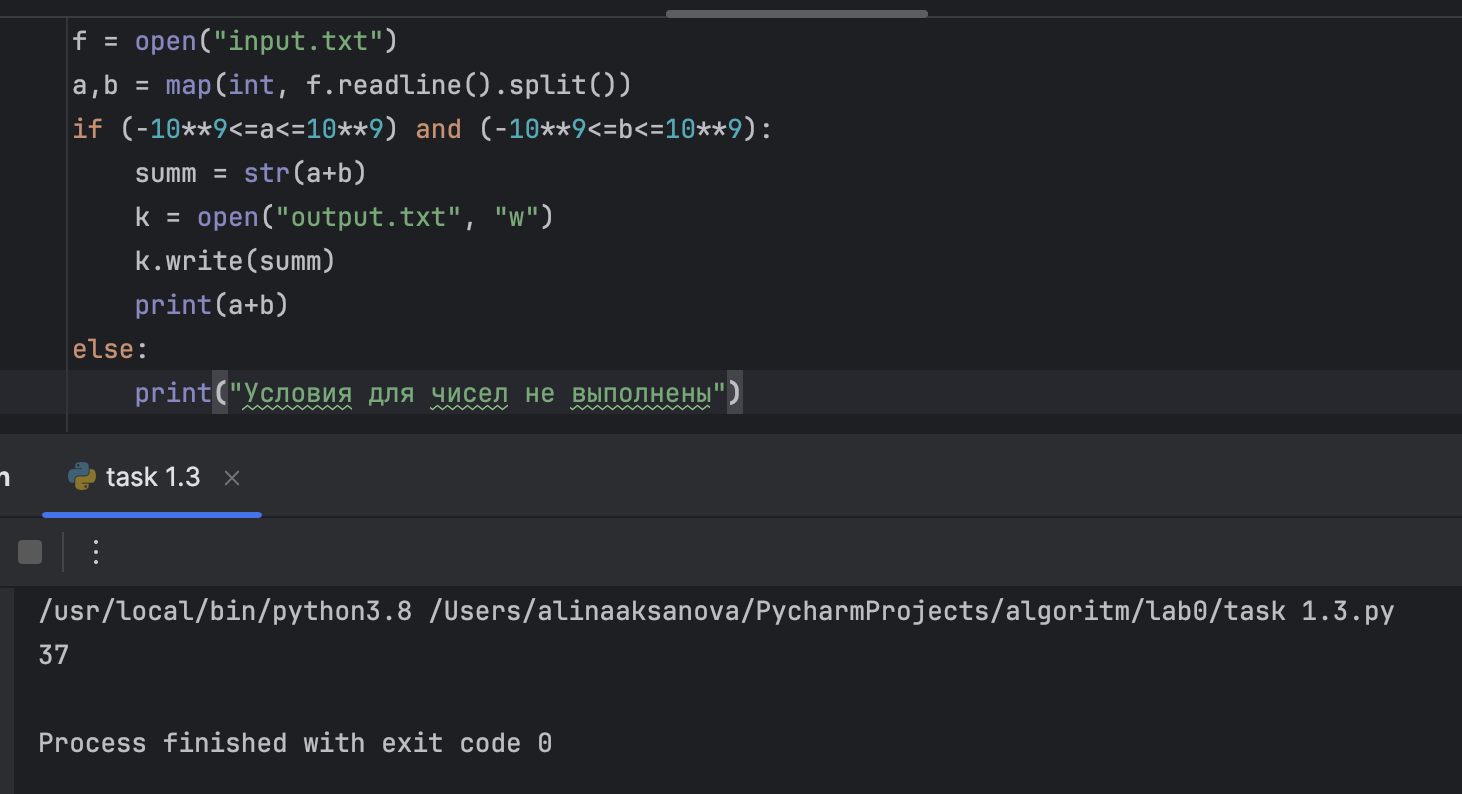
Сначала я открываю файл input.txt и считываю строчку с 2 значениями в нем. Далее провожу проверку на соответствие чисел с условиями. Если все верно, я вношу в переменную sum сумму чисел, открываю файл  output.txt и заношу туда значение переменной summ, после чего вывожу это значение на экран. Если же числа не удовлетворяют условиям, вывожу “Условия для чисел не выполнены”.

1. Результат работы кода на примерах из текста задачи:

Файлы input.txt и output.txt:



 Вывод результата:



1. Вывод

Я вспомнила принцип работы с файлами, способы их чтения и записи данных в них.

### Задача 1.4

1. Текст задачи

Выполните задачу a+b2 c использованием файлов аналогично предыдущему пункту.

1. Листинг кода

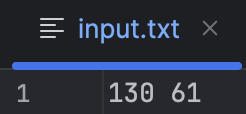
f = open("input.txt")  
a,b = map(int, f.readline().split())  
if (-10\*\*9<=a<=10\*\*9) and (-10\*\*9<=b<=10\*\*9):  
 summ = str(a+b\*b)  
 k = open("output.txt", "w")  
 k.write(summ)  
 print(summ)  
else:  
 print("Условия для чисел не выполнены")

1. Текстовое объяснение решения.

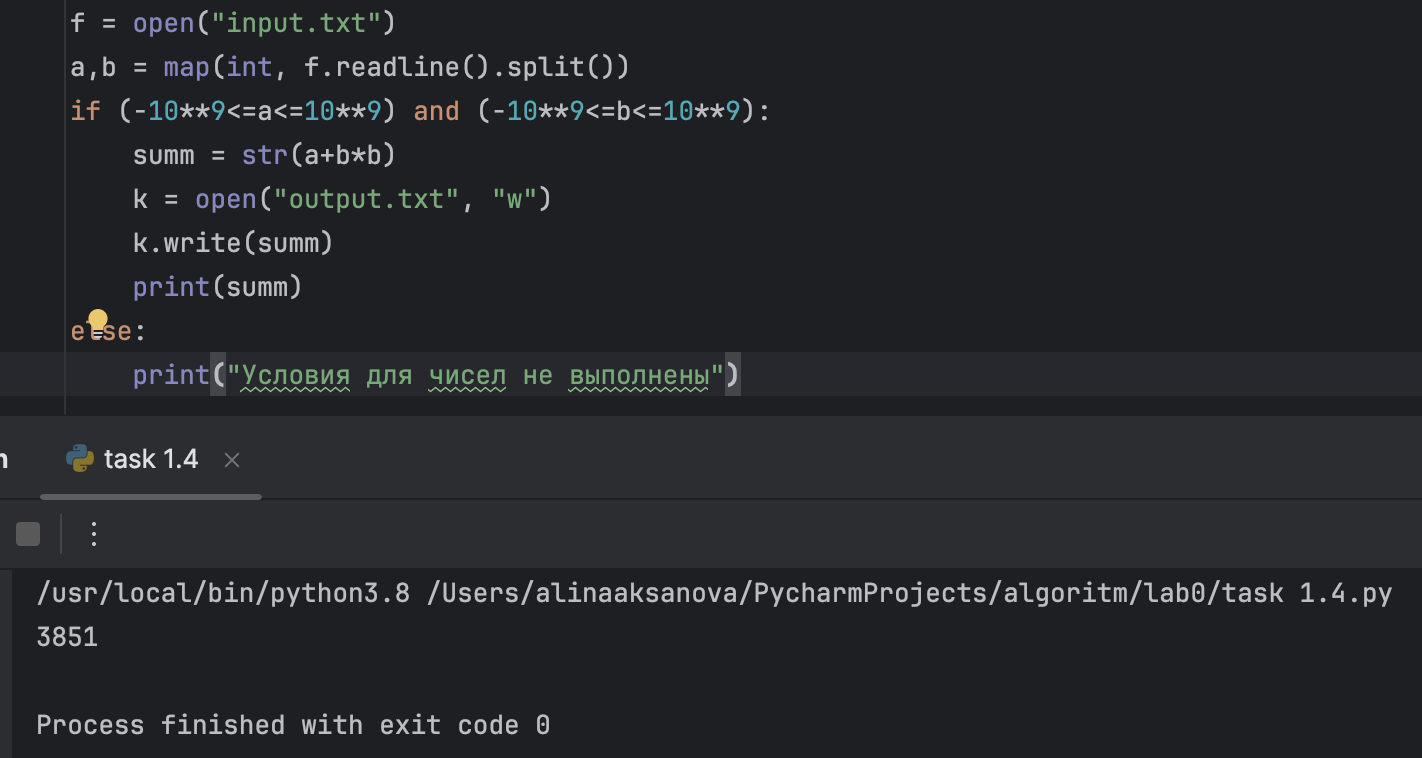
Задача решается аналогично задаче 1.3, за исключением того, что в переменную sum мы записываем сумму числа a и квадрата числа b.

1. Результат работы кода на примерах из текста задачи:

Файлы input.txt и output.txt:



Вывод результата:



1. Вывод по задаче:

Вывод аналогичен выводу в задаче 1.3, а именно: Я вспомнила принцип работы с файлами, способы их чтения и записи данных в них

## Задача №2. Число Фибоначчи

1. Текст задачи

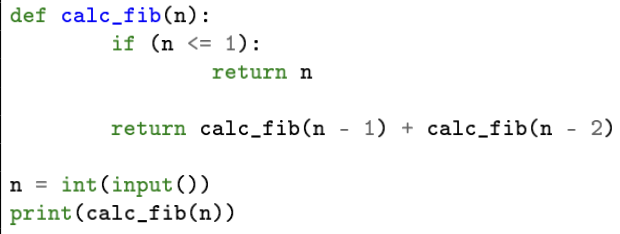
Определение последовательности Фибоначчи:

F0 = 0

F1 = 1

Fi = Fi − 1 + Fi – 2 для i ≥ 2.  
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: 

• Имя входного файла: input.txt

• Имя выходного файла: output.txt

• Формат входного файла. Целое число n. 0 ≤ n ≤ 45.

• Формат выходного файла. Число Fn.

• Пример

|  |  |
| --- | --- |
| input.txt | 10 |
| output.txt | 55 |

1. Листинг кода

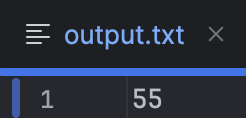
k = open("input.txt")  
n = int(k.readline())  
f1,f2 = 0,1  
for i in range(2,n+1):  
 f1,f2=f2, (f1+f2)  
y = open("output.txt", "w")  
y.write(str(f2))  
print(f2)

1. Текстовое объяснение решения.

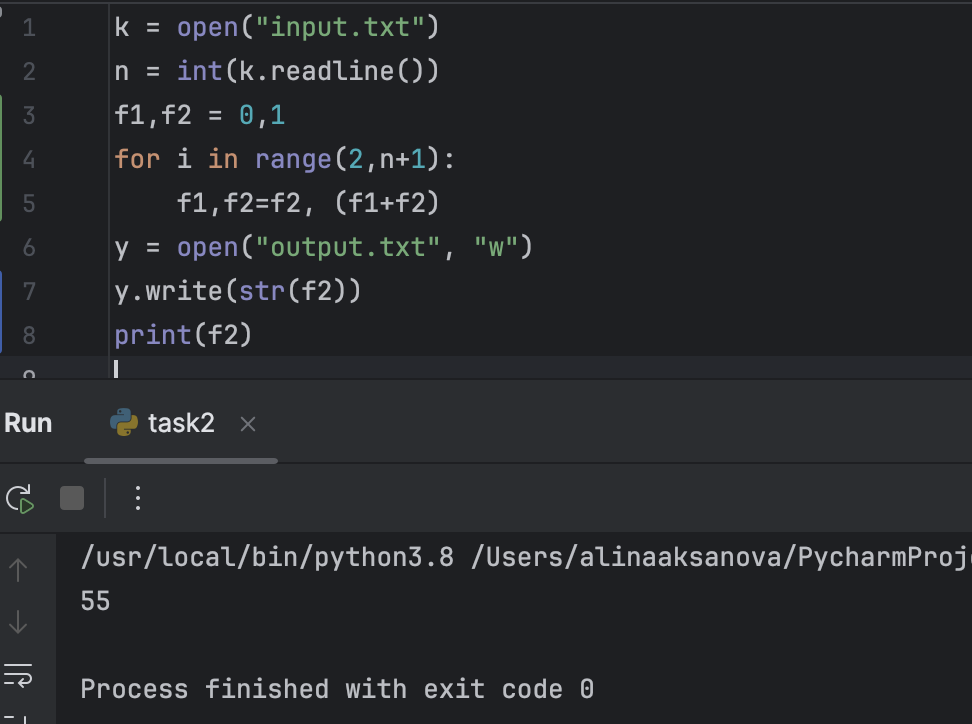
Считав числа из файла input.txt, ввожу 2 переменные f1 и f2 (f2 является следующим числом последовательности после f1), с помощью которых буду подсчитывать число Фибоначчи. Циклом высчитываю поочереди числа f1 и f2 и обновляю их значения. Записываю конечное число Фибоначчи в файл output.txt , а так же вывожу на экран.

1. Результат работы кода на примерах из текста задачи:

Файлы input.txt и output.txt:



Вывод результата:



1. Вывод по задаче:

Я написала алгоритм для подсчета чисел Фибоначчи, а так же я поработала с файлами, со считыванием информации из них и ее записи в файл.

## Задача №3. Еще про числа Фибоначчи

1. Текст задачи

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

F200 = 280571172992510140037611932413038677189525

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

* Имя входного файла: input.txt
* Имя выходного файла: output.txt
* Формат входного файла. Целое число n. 0 ≤ n ≤ 107.
* Формат выходного файла. Одна последняя цифра числа Fn.
* Пример 1.

|  |  |
| --- | --- |
| input.txt | 331 |
| output.txt | 9 |

F331 = 668996615388005031531000081241745415306766517246774551964595292186469.

* Пример 2.

|  |  |
| --- | --- |
| input.txt | 327305 |
| output.txt | 5 |

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

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

1. Листинг кода

f = open("input.txt")  
n = int(f.readline())  
f1,f2 = 0,1  
for i in range(2,n+1):  
 f1,f2=f2, (f1+f2)%10  
k = open("output.txt", "w")  
k.write(str(f2))  
print(f2)

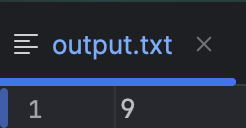
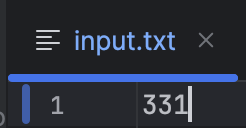
1. Текстовое объяснение решения.

Я считала данные из файла input.txt, далее ввела 2 переменные f1 и f2, для подсчета числа Фибоначчи (f2 является следующим числом последовательности после f1). Далее с помощью цикла я перебираю числа Фибоначчи поочереди, обновляя значение переменных f1 и f2. Когда цикл завершается, я вношу в файл output.txt последнюю цифру числа. В конце вывожу результат.

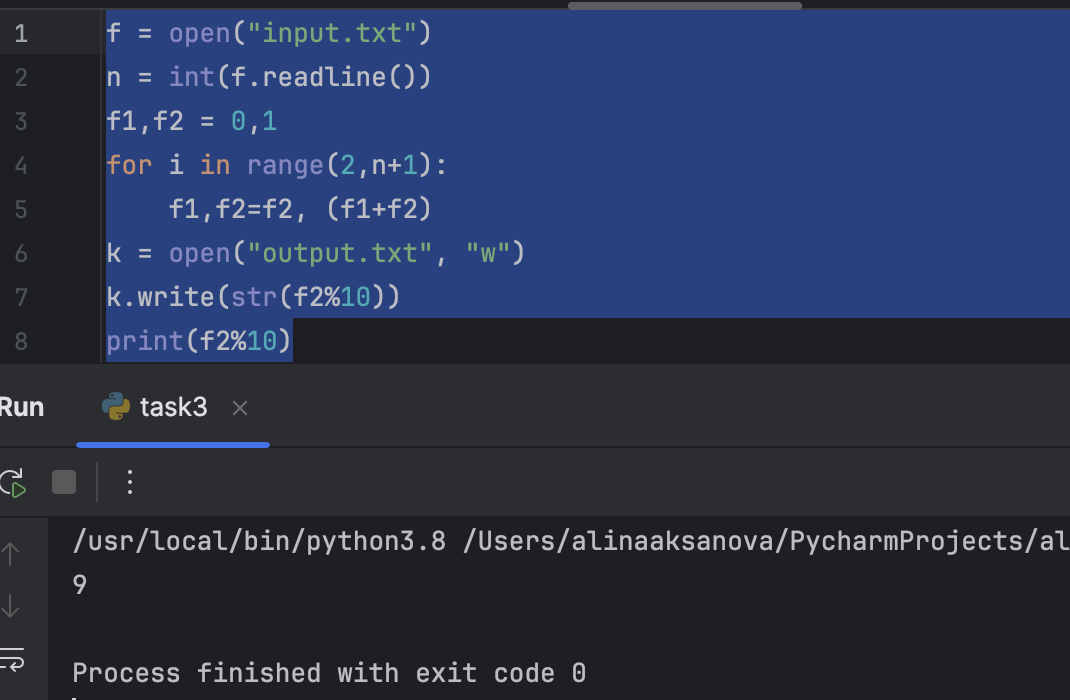
1. Результат работы кода на примерах из текста задачи:

Пример 1:

Файлы input.txt и output.txt:

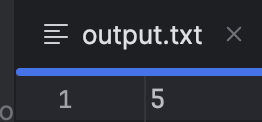
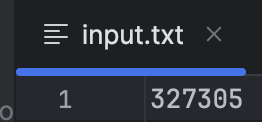


Вывод результата:

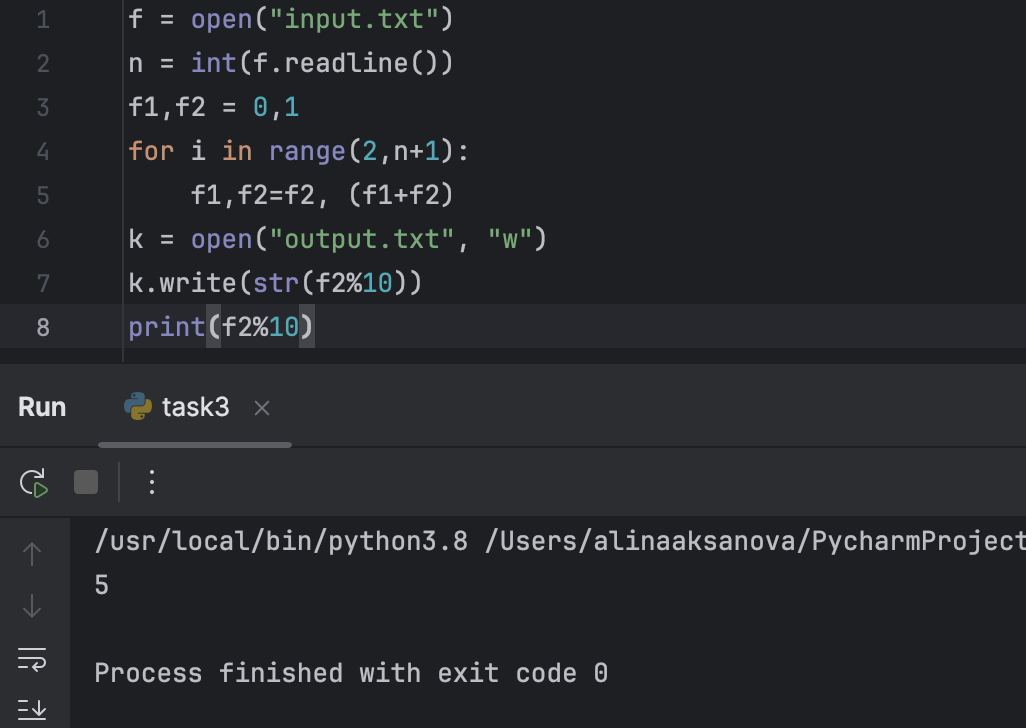


Пример 2:

Файлы input.txt и output.txt:



Вывод результата:



1. Вывод по задаче:

Я разработала алгоритм, который высчитывает число Фибоначчи и выводит его последнюю цифру , а так же провела работу с файлами

## Задача №4. Тестирование ваших алгоритмов

1. Текст задачи

вам необходимо протестировать время выполнения вашего алгоритма в *Задании 2* и *Задании 3*.

### Задача 4.1(тест задачи 2)

1. Листинг кода

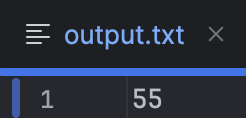
import time  
t\_start = time.perf\_counter()  
k = open("input.txt")  
n = int(k.readline())  
f1,f2 = 0,1  
for i in range(2,n+1):  
 f1,f2 = f2, (f1+f2)  
y = open("output.txt", "w")  
y.write(str(f2))  
print(f2)  
print("Время работы: %s секунд" % (time.perf\_counter() - t\_start))

1. Текстовое объяснение решения.

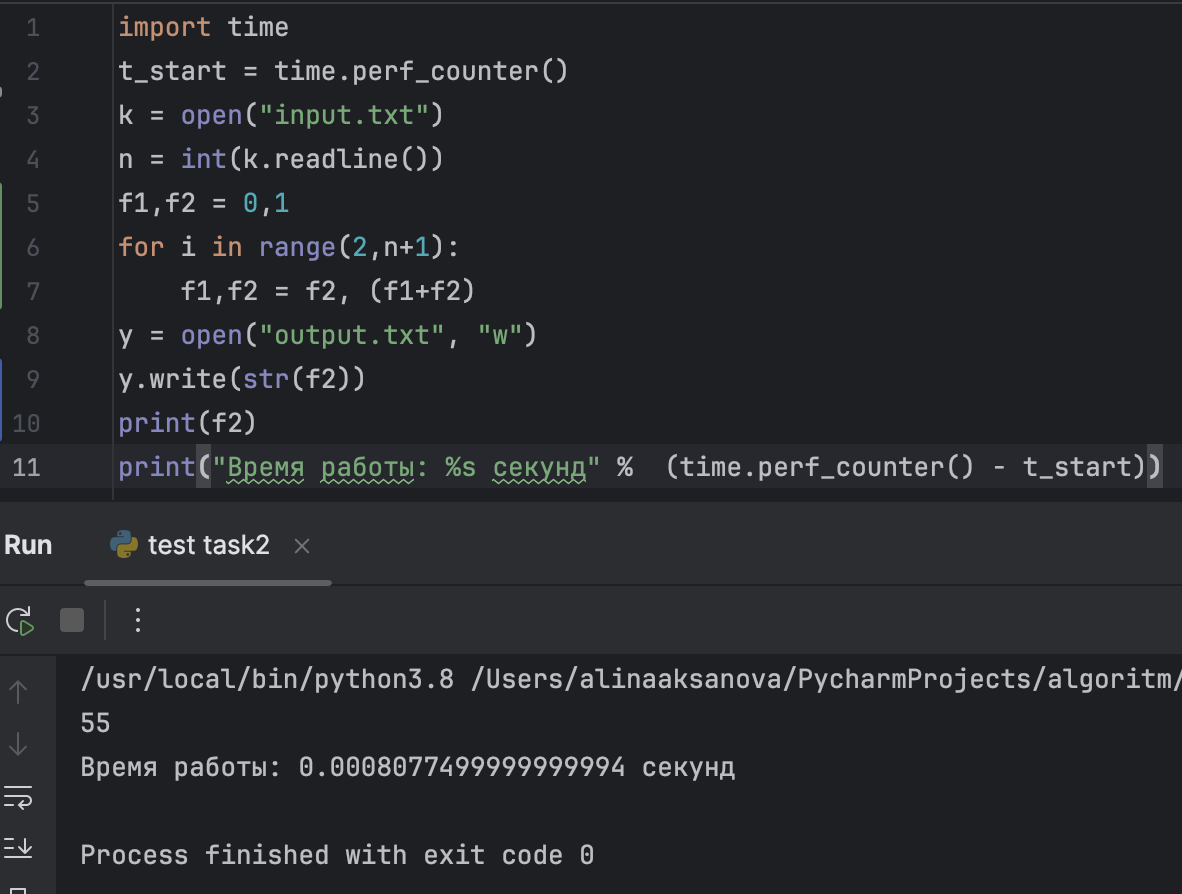
С помощью предложенного в задаче кода я дополняю свой код из задания 2, и измеряю время работы кода.

1. Результат работы кода на примерах из текста задачи:

Файлы input.txt и output.txt:

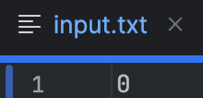
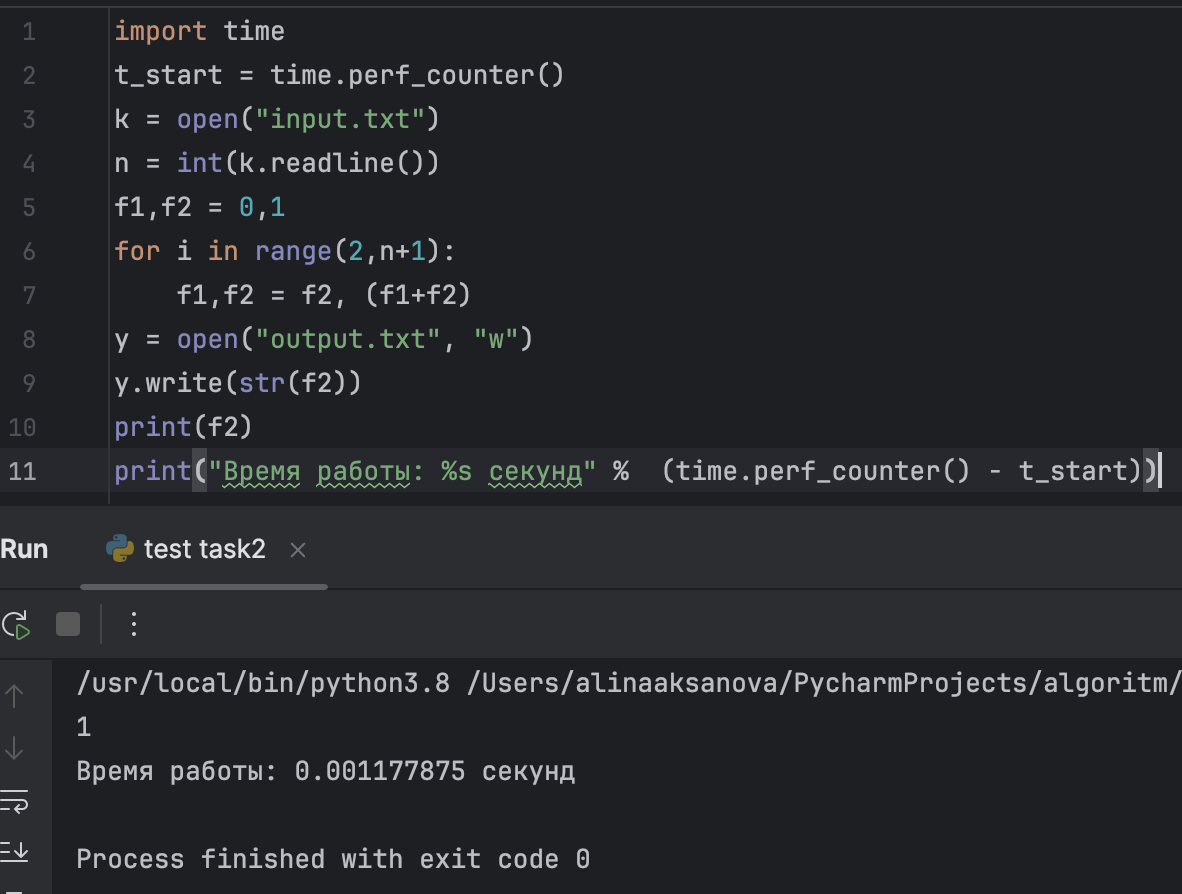


Вывод результата:

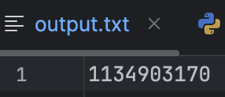
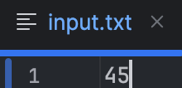
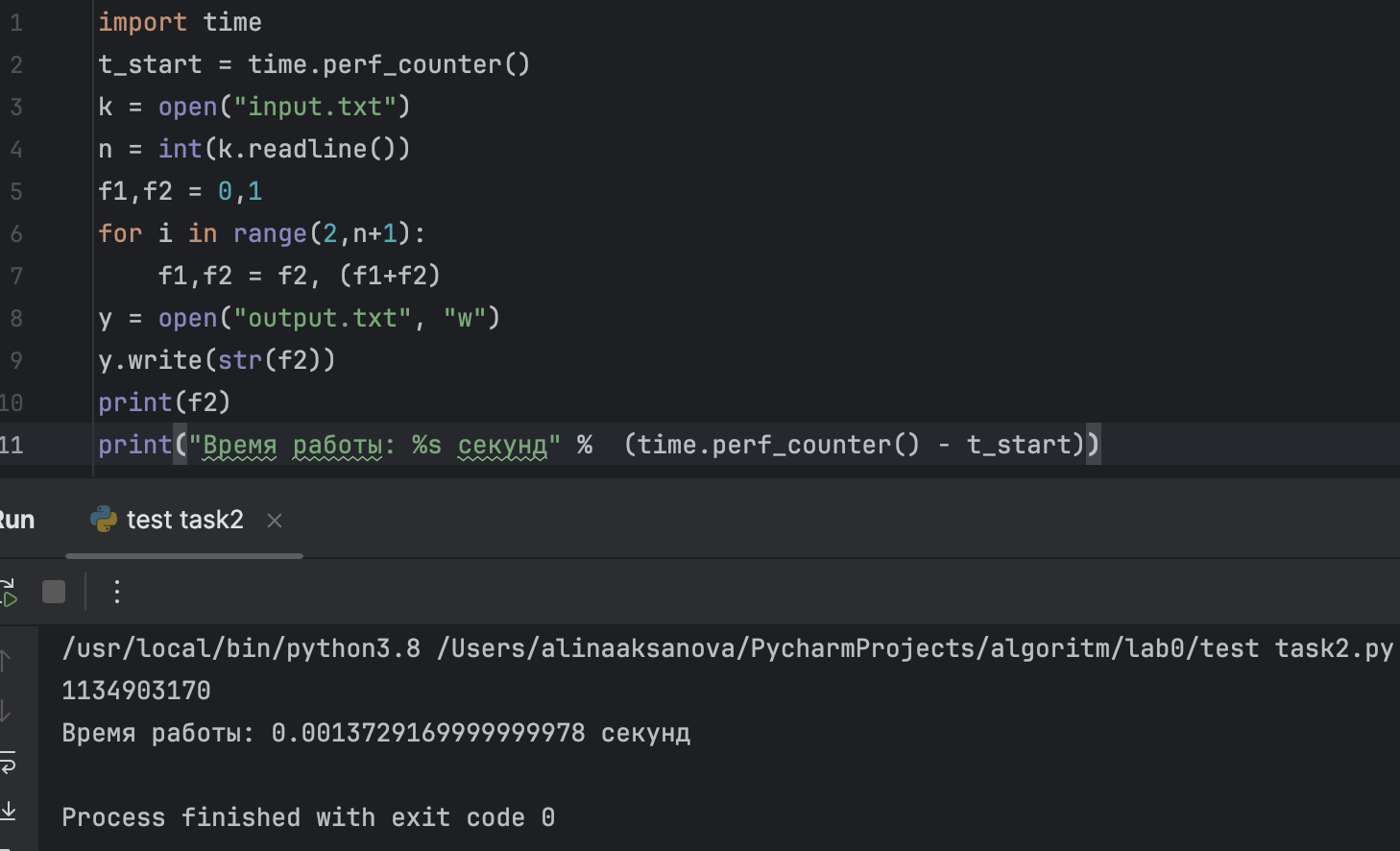


Результат работы кода на максимальных и минимальных значениях:

Min:



Max:



1. Вывод по задаче:

Замерив время работы алгоритма, я поняла что он работает быстро как на маленьких, так и на больших значениях.

### Задача 4.2 ( тест задачи 3)

1. Листинг кода

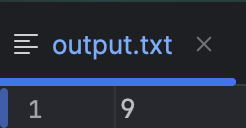
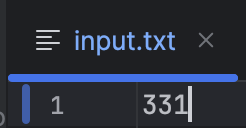
import time  
t\_start = time.perf\_counter()  
f = open("input.txt")  
n = int(f.readline())  
f1,f2 = 0,1  
for i in range(2,n+1):  
 f1,f2 = f2, (f1+f2)%10  
open("output.txt", "w").write(str(f2))  
print(f2)  
print("Время работы: %s секунд" % (time.perf\_counter() - t\_start))

1. Текстовое объяснение решения.

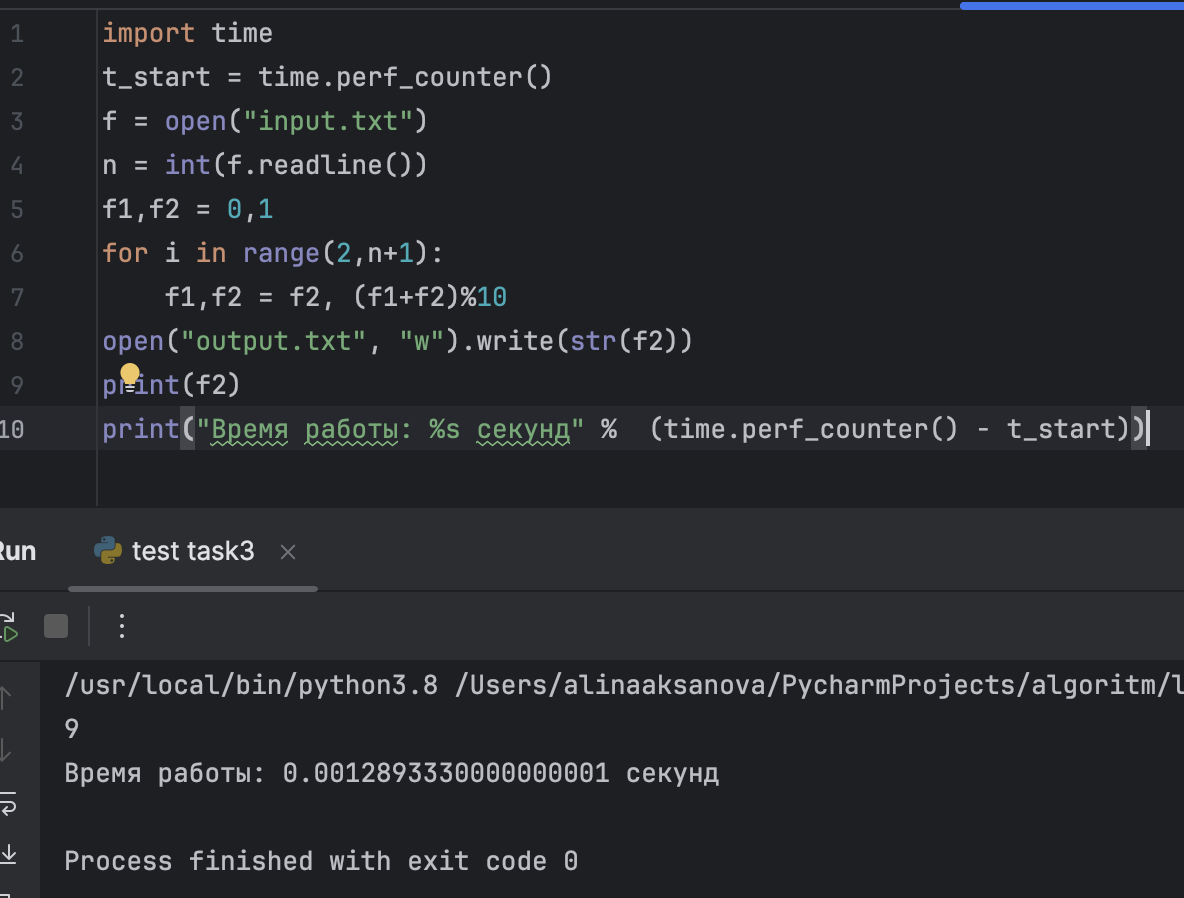
Аналонично задаче 4.1, с помощью предложенного в задаче кода я дополняю свой код из задания 3, и измеряю время работы кода.

1. Результат работы кода на примерах из текста задачи:

Файлы input.txt и output.txt:

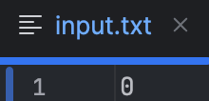
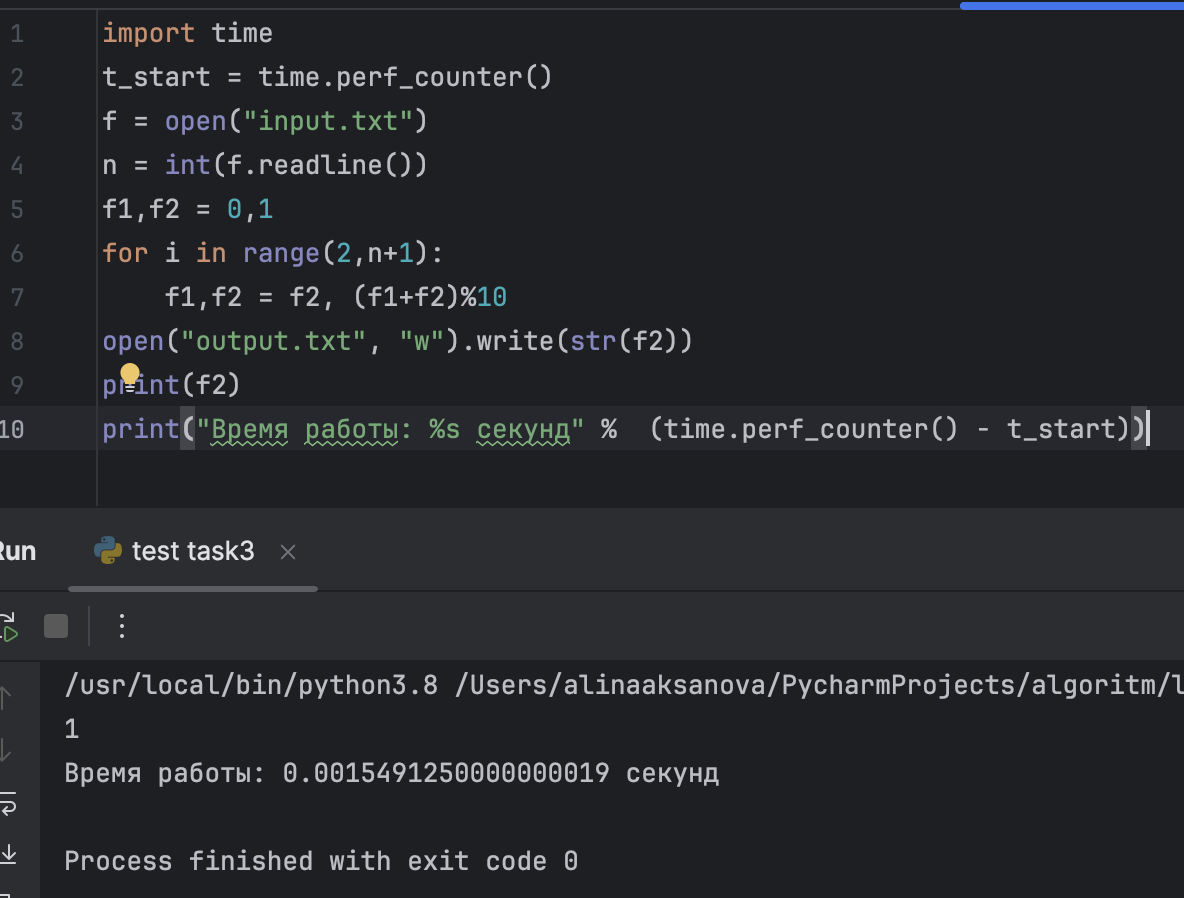


Вывод результата:

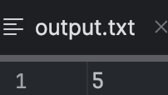
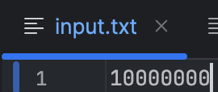
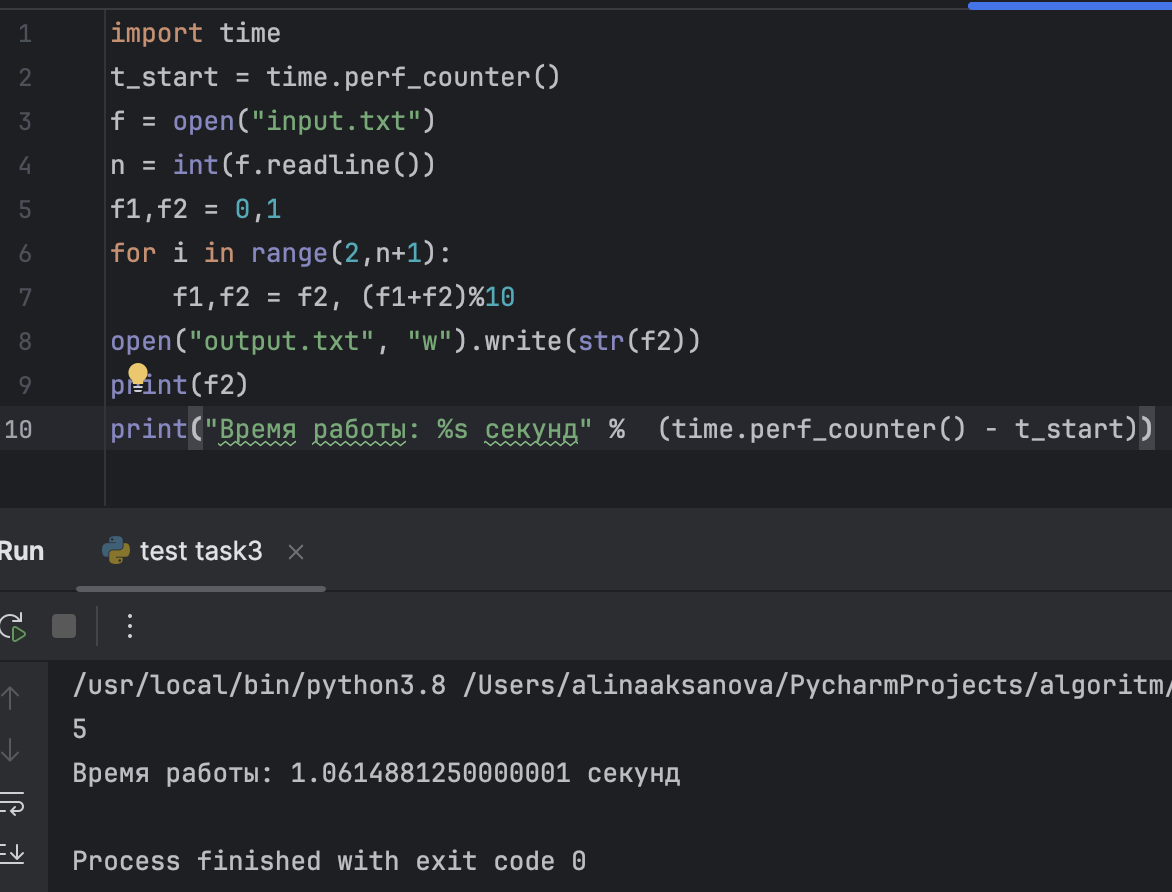


Результат работы кода на максимальных и минимальных значениях:

Min:



Max:



1. Вывод по задаче:

Замерив время работы алгоритма , я поняла что он работает быстро как на маленьких, так и на больших значениях.

# Вывод

В результате лабораторной я вспомнила как писать базовый код на Python, как работать с файлами, считывать из них информацию и записывать ее туда. Также я разработала алгоритм для подсчета числа Фибоначчи и вывода его последнего числа. В заключении я научилась замерять время работы алгоритма и выяснила, что написанные мною коды работают эффективно как на малых значениях, так и на больших.