

--	--	--

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

Отчет по лабораторной работе №1
по курсу «Алгоритмы и структуры данных»
Тема: Введение. Работа с файлами, тестирование.
Вариант 1

Выполнила:
Толстухина К.А.
К3139

Проверила:
Афанасьев А.В.

--	--	--

--	--	--

Санкт-Петербург
2024 г.
Содержание отчета

Содержание отчета	2
Задачи по варианту:	
Задание №1 Ввод-вывод.	3
Задание №2 Число Фибоначчи.	10
Задание №3 Еще про числа Фибоначчи.	12
Задание №4 Тестирование ваших алгоритмов.	13
Вывод	15

		2
--	--	---

--	--	--

Задачи по варианту

Задание №1. Ввод-вывод

Задача 1: Задача a + b. В данной задаче требуется вычислить сумму двух заданных чисел. Вход: одна строка, которая содержит два целых числа a и b. Выход: единственное целое число - результат сложения a + b.

```
a, b = map(int, input().split())
f = True
while f:
    if -10 ** 9 <= a <= 10 ** 9 and -10 ** 9 <= b <= 10 ** 9:
        f = False
        break
    print('Некорректные данные. Введите еще раз:')
    a, b = map(int, input().split())
print(a + b)
```

Так как два вводных числа находятся в одной строке - я использую `split()`, чтобы разбить строку на список с двумя необходимыми числами (которые по умолчанию являются строкой) и далее через встроенную функцию `map` “перевожу” `str` формат в `int` и соответственно присваиваю двум переменным `a` и `b` два получившихся числа. Далее вывожу сумму данных чисел.

		3
--	--	---

--	--	--

Примеры.

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

Результат работы кода:

```

Run 1 x
/Users/kseniato1stuhina
/PycharmProjects
/pythonProject/
.venv/bin/python
/Users/kseniato1stuhina
/PycharmProjects
/pythonProject/1.py
12 25
37

Process finished with exit
code 0

```

```

Run 1 x
/Users/kseniato1stuhina
/PycharmProjects
/pythonProject/
.venv/bin/python
/Users/kseniato1stuhina
/PycharmProjects
/pythonProject/1.py
130 61
191

Process finished with exit
code 0

```

Результат работы кода на максимальном значении:

--	--	--

```

Run 1 x
/Users/kseniato1stuhina
/PycharmProjects
/pythonProject/
.venv/bin/python
/Users/kseniato1stuhina
/PycharmProjects
/pythonProject/1.py
1000000000 1000000000
2000000000

Process finished with exit
code 0

```

Результат работы кода на минимальном значении:

```

Run 1 x
/Users/kseniato1stuhina
/PycharmProjects
/pythonProject/
.venv/bin/python
/Users/kseniato1stuhina
/PycharmProjects
/pythonProject/1.py
-1000000000 -1000000000
-2000000000

Process finished with exit
code 0

```

Вывод по задаче: код работает, я вспомнила базовые методы и функции в Питоне.

Задача 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 . Выход: единственное целое число — результат сложения $a + b^2$.

		5
--	--	---

```

a, b = map(int, input().split())
f = True
while f:
    if -10 ** 9 <= a <= 10 ** 9 and -10 ** 9 <= b <= 10 ** 9:
        f = False
        break
    print('Некорректные данные. Введите еще раз:')
    a, b = map(int, input().split())
print(a + b ** 2)

```

Так как два вводных числа находятся в одной строке - я использую `split()`, чтобы разбить строку на список с двумя необходимыми числами (которые по умолчанию являются строкой) и далее через встроенную функцию `map` “перевожу” `str` формат в `int` и соответственно присваиваю двум переменным `a` и `b` два получившихся числа. Далее вывожу сумму первого числа с квадратом второго, для возведения в квадрат я использовала оператор `**`.

Примеры:

The screenshot shows the PyCharm Run console. The file path is `/Users/kseniato1stuhina/PycharmProjects/pythonProject/.venv/bin/python /Users/kseniato1stuhina/PycharmProjects/pythonProject/2.py`. The input is `12 25` and the output is `637`. The process finished with exit code 0.

The screenshot shows the PyCharm Run console. The file path is `/Users/kseniato1stuhina/PycharmProjects/pythonProject/.venv/bin/python /Users/kseniato1stuhina/PycharmProjects/pythonProject/2.py`. The input is `130 61` and the output is `3851`. The process finished with exit code 0.

Результат на максимальном значении:

--	--	--

```

/Users/kseniato1stuhina
/PycharmProjects
/pythonProject/
.venv/bin/python
/Users/kseniato1stuhina
/PycharmProjects
/pythonProject/2.py
1000000000 1000000000
10000000001000000000

Process finished with exit
code 0

```

Результат на минимальном значении:

```

/Users/kseniato1stuhina
/PycharmProjects
/pythonProject/
.venv/bin/python
/Users/kseniato1stuhina
/PycharmProjects
/pythonProject/2.py
-10000000000 -10000000000
999999999000000000

Process finished with exit
code 0

```

Вывод по задаче: вспомнила оператор в Питоне

Задача 3. Выполните задачу $a + b$ с использованием файлов. Имя входного файла: input.txt, имя выходного файла: output.txt. Входной файл состоит из одной строки, которая содержит a и b . Выходной файл единственное целое число - результат сложения $a + b$.

		7
--	--	---

--	--	--

```

with open('input.txt', 'r') as f:
    a, b = map(int, f.readline().split())
    flag = True
    while f:
        if -10 ** 9 <= a <= 10 ** 9 and -10 ** 9 <= b <= 10 **
9:
            flag = False
            break
        print('Некорректные данные. Введите еще раз:')
        a, b = map(int, input().split())

f.close()

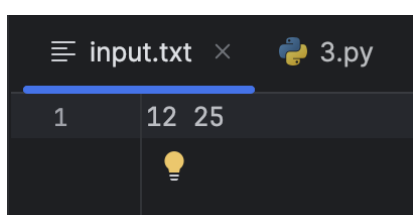
with open('output.txt', 'w') as f:
    f.write(str(a + b))

f.close()

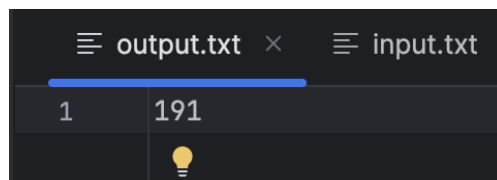
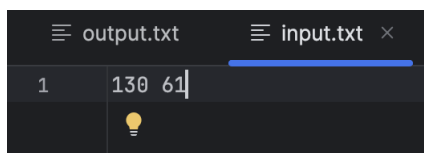
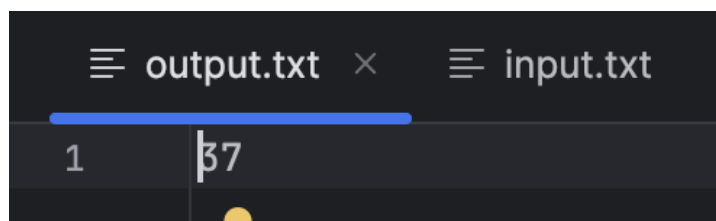
```

С использованием оператора with открываю файл и считываю данные через функцию readline(), также через функцию map перевожу считываемые данные в числовой формат и присваиваю переменным a и b соответственно. Потом также открываю файл, где должен быть записан результат и через метод write записываю результат сложения в str формате.

Данные в файле input.txt:



Данные в выходном файле:



Результат на минимальном значении:

--	--	--

☰ input.txt ×	
1	-10000000000 -10000000000
☰ output.txt ×	
1	-20000000000

Результат на максимальном значении:

☰ input.txt ×	
1	10000000000 10000000000
☰ output.txt ×	
1	20000000000

Вывод по задаче: я вспомнила, как работать с файлами, считывать и загружать результирующие данные.

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

```
with open('input.txt', 'r') as f:
    a, b = map(int, f.readline().split())
    flag = True
    while f:
        if -10 ** 9 <= a <= 10 ** 9 and -10 ** 9 <= b <= 10 **
9:
            flag = False
            break
        print('Некорректные данные. Введите еще раз:')
        a, b = map(int, input().split())

f.close()

with open('output.txt', 'w') as f:
```

		9
--	--	---

--	--	--

```
f.write(str(a + b ** 2))

f.close()
```

Принцип работы аналогичен предыдущей задаче за исключением тех данных, которые я записываю в файл (к а прибавляется квадрат числа b).

Результаты работы программы, входные и выходные данные в файле:

output.txt		input.txt	
1	130	61	

output.txt		input.txt	
1	3851		

input.txt		3.py	
1	12	25	

output.txt		input.txt	
1	37		

Результат на минимальном значении:

input.txt			
1	-10000000000	-10000000000	
output.txt			
1	9999999990000000000		

Результат на максимальном значении:

		10
--	--	----

--	--	--

☰ input.txt ×	
1	1000000000 1000000000
☰ output.txt ×	
1	10000000001000000000

Вывод по задаче: я вспомнила, как работать с файлами, считывать и загружать результирующие данные.

Задание №2. Число Фибоначчи.

Ваша цель - разработать эффективный алгоритм для подсчета чисел Фибоначчи. Вам предлагается начальный код на Python, который содержит наивный рекурсивный алгоритм.

```
n = int(open('input.txt').readline())
f = True
while f:
    if 0 <= n <= 45:
        f = False
        break
    print('Неправильное значение n. Введите еще раз:')
    n = int(input())

first, second = 0, 1
for i in range(int(n) - 1):
    new = first + second
    first = second
```

		11
--	--	----

--	--	--

```

second = new
if n <= 1:
    new = n
with open('output.txt', 'w') as f:
    f.write(str(new))
f.close()

```

Открываю файл и считываю данные, перевожу в числовой формат, так как далее мне предстоит работать с числами. Использую базовый алгоритм подсчета числа Фибоначчи через три переменные, со сдвигающимися значениями. Полученное число записываю в файл.

Результат работы:

output.txt		input.txt	
1	10	1	55

Результат на минимальном значении:

input.txt	
1	0
output.txt	
1	0

Результат на максимальном значении:

--	--	--

☰ input.txt ×	
1	45
☰ output.txt ×	
1	1134903170

Вывод по задаче: вспомнила один из легких путей подсчета числа Фибоначчи, так же продолжаю практиковать работу с файлами.

--	--	--

Задание №3. Еще про числа Фибоначчи.

Определение последней цифры большого числа Фибоначчи.

```
n = int(open('input.txt').readline())
f = True
while f:
    if 0 <= n <= 10 ** 7:
        f = False
        break
    print('Неправильное значение n. Введите еще раз:')
    n = int(input())

first, second = 0, 1
for i in range(int(n) - 1):
    new = first + second
    first = second
    second = new
if n <= 1:
    new = n

with open('output.txt', 'w') as f:
    f.write(str(new % 10))
f.close()
```

Используя тот же алгоритм высчитываю число и в результат записывая остаток от деления числа на 10, т.е. последнюю цифру получившегося числа.

Результаты тестов:

output.txt	input.txt
1	331

output.txt	input.txt
1	327305

output.txt	input.txt
1	5

Что соответствует правильным ответам:

		14
--	--	----

--	--	--

- 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.

Вывод по задаче: вспомнила, как быстро находить последнюю цифру числа, даже если оно очень большое.

Задание №4. Тестирование ваших алгоритмов.

Необходимо протестировать время выполнения вашего алгоритма в задании №2 и №3

```
import time
t_start = time.perf_counter()

n = int(open('input.txt').readline())
f = True
while f:
    if 0 <= n <= 45:
        f = False
        break
    print('Неправильное значение n. Введите еще раз:')
    n = int(input())

first, second = 0, 1
for i in range(int(n) - 1):
    new = first + second
    first = second
    second = new
if n <= 1:
    new = n

with open('output.txt', 'w') as f:
    f.write(str(new))
f.close()

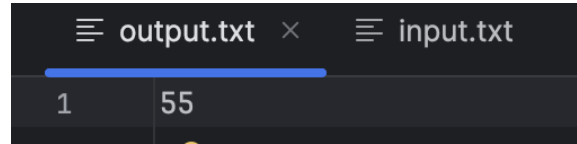
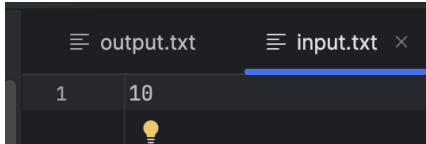
print(time.perf_counter() - t_start)
```

Я импортирую библиотеку, чтобы отследить время выполнения кода. Засаекаю время начала работы кода и записываю в переменную t_start. Далее копирую

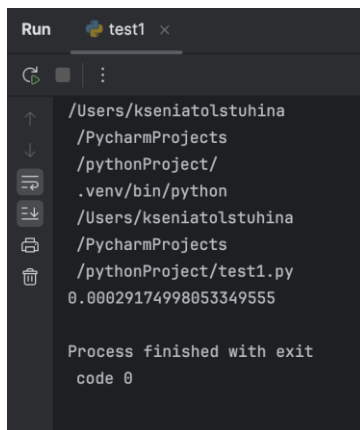
--	--	--

код, написанный ранее. И вывожу общее время выполнения через разницу текущего с начальным.

Результаты теста:



Затраченное время:



Вывод по задаче: выполнение программы укладывается во время, также я вспомнила как работать с модулем time.

Так же код для второй задачи (задачи №3):

```
import time

t_start = time.perf_counter()

n = int(open('input.txt').readline())
f = True
while f:
    if 0 <= n <= 10 ** 7:
        f = False
        break
    print('Неправильное значение n. Введите еще раз:')
    n = int(input())

first, second = 0, 1
for i in range(int(n) - 1):
    new = first + second
    first = second
    second = new
if n <= 1:
    new = n
```


--	--	--

```
with open('output.txt', 'w') as f:
    f.write(str(new % 10))
f.close()

print(time.perf_counter() - t_start)
```

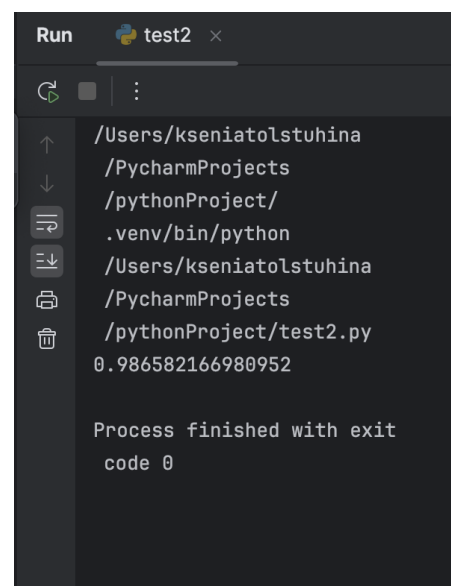
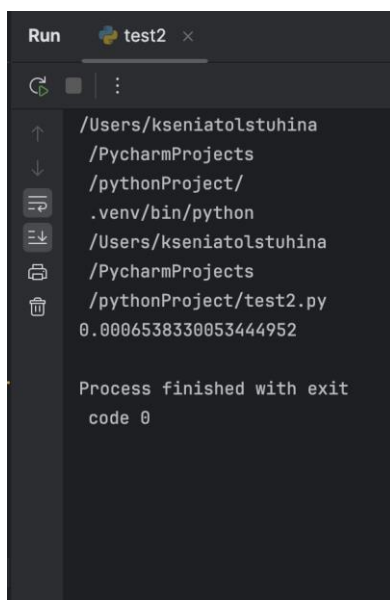
Я делаю все тоже самое, только изменены данные, записываемые в результирующий файл.

Так как основная часть кода не поменялась, то на примерах все работает точно так же, можно посмотреть в Задании №3

Время, затраченное на первый и второй тест соответственно:

Первый тест с входными данными: 331

Второй тест с входными данными: 327305



Вывод по задаче: с большими данными так же укладывается вовремя.

Вывод по лабораторной работе: я вспомнила многие функции, модули, операторы в Питоне, а также попрактиковала работу с файлами.