

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:

Криличевский М. Е.

Номер группы: К3139

Проверил:

Афанасьев А. В.

Санкт-Петербург

2024 г.

Оглавление

Задания.....	3
Задание 1.....	3
1. Задача $a + b$	3
2. Задача $a + b^{**}2$	3
3. Задача $a+b$ с файлом.....	4
4. Задача $a+b$ с файлом.....	5
Задание 2. Число Фибоначчи.....	6
Задание 3. Последняя цифра Фибоначчи.....	7
Задание 4. Анализ времени работы программ в задании 2 и 3.....	8
Тестирование кода в задании 2.....	8
Тестирование кода в задании 3.....	9
Вывод:.....	10

Задания

Задание 1.

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 $a + b$.

```
1 a, b = map(int, input().split())
2 if -10**9<=a<=10**9 and -10**9<=b<=10**9: print(a+b)
3 else: print("Ошибка")
```

Объяснение решения: на консоли 2 числа, в пределах диапазона, записываются через пробел. Если два числа входят в указанный в условии диапазон, то программа выводит сумму данных чисел. Иначе программа выводит ошибку.

Пример работы программы:

```
/usr/local/bin/python3.12
8 9
17
Process finished with exit
```

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 . Для этих чисел выполняются условия $-10^9 \leq a, b \leq 10^9$. Выход: единственное целое число — результат сложения $a + b^2$.

```
1 a, b = map(int, input().split())
2 if -10**9<=a<=10**9 and -10**9<=b<=10**9: print(a + b ** 2)
3 else: print("Ошибка")
```

Объяснение решения: на консоли 2 числа, в пределах диапазона, записываются через пробел. Если два числа входят в указанный в условии диапазон, то программа выводит значение выражения. Иначе программа выводит ошибку.

Пример работы программы:

```
/usr/local/bin/py
7 5
32

Process finished
```

3. Задача $a+b$ с файлом. 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 $a + b$.

```
1 with open("input.txt", "r") as f:
2     for s in f:
3         a1 = [int(x) for x in s.split()]
4         a,b = a1[0],a1[1]
5         t = a+b
6
7     with open("output.txt", "w") as z:
8         if -10**9<=a<=10**9 and -10**9<=b<=10**9:
9             z.write(str(t))
10        else: z.write("Ошибка")
11    z.close()
```

Объяснение решения: программа открывает и читает файл input.txt с двумя числами, записанными через пробел. С помощью цикла for программа перебирает строку с числами и отправляет данные числа в список. Переменным a и b присваивается значение элементов списка. Переменная t обозначает сумму двух элементов списка. Затем программа открывает файл output.txt. Если оба числа входят в заданный в условии диапазон, то программа записывает в файл output.txt строковый тип числа t , которое является суммой двух чисел в файле input.txt, и закрывает файл output.txt. Иначе программа записывает в файл output.txt слово «Ошибка» и закрывает файл output.txt.

Пример работы программы:

1	12 25
2	
3	

1	37
2	
3	

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

```

1  with open("input.txt", "r") as f:
2      for s in f:
3          a1 = [int(x) for x in s.split()]
4          a,b = a1[0],a1[1]
5          t = a+b**2
6
7  with open("output.txt", "w") as z:
8      if -10**9<=a<=10**9 and -10**9<=b<=10**9:
9          z.write(str(t))
10     else: z.write("Ошибка")
11     z.close()

```

Объяснение решения: программа открывает и читает файл input.txt с двумя числами, записанными через пробел. С помощью цикла for программа перебирает строку с числами и отправляет данные числа в список. Переменным a и b присваивается значение элементов списка. Переменная t обозначает сумму двух элементов списка. Затем программа открывает файл output.txt. Если оба числа входят в заданный в условии диапазон, то программа записывает в файл output.txt строковый тип числа t, которое является значением выражения, и закрывает файл output.txt. Иначе программа записывает в файл output.txt слово «Ошибка» и закрывает файл output.txt.

Пример работы программы:

1	130 61
2	
3	

1	191
2	
3	

Задание 2. Число Фибоначчи

Определение последовательности Фибоначчи:

- 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

```
1  with open("fib_input", "r") as f:
2      for s in f:
3          a = [int(x) for x in s.split()]
4          n = a[0]
5
6      fb1 = fb2 = 1
7      for i in range(2,n):
8          fb1,fb2 = fb2,fb1+fb2
9      t = str(fb2)
10
11  with open("fib_output", "w") as z:
12      if 0<=n<=45: z.write(str(t))
13      else: print("Ошибка")
14      z.close()
```

Объяснение решения: программа открывает и читает файл fib_input.txt с числом. С помощью цикла for программа перебирает строку с единственным числом и отправляет его в список. Переменной n-единственный элемент списка. Затем, после создания переменных с первыми двумя единицами последовательности цифр числа Фибоначчи, с помощью цикла for, который перебирает значения от 2 до значения переменной n. Далее переменной fb1 присваивается значение fb2, а переменной значение суммы fb1 и fb2. Переменной t присвоено строковое значение суммы fb1 и fb2. После этого программа открывает файл output.txt для потенциального добавления результата. Если число n находится в заданном диапазоне, то в файл output.txt добавляется значение переменной t. Иначе программа выводит ошибку.

Пример работы программы:

Цифра во входном файле:

1	0
2	
3	

Цифра в выходном файле:

1	1
2	
3	

Задание 3. Последняя цифра 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 .

```

1  def LF(x): 1 usage ± KrimaxeV *
2      if x==0: return 1
3      fb1,fb2 = 0,1
4      for i in range(x-1):
5          fb = (fb1+fb2)%10
6          fb1,fb2 = fb2,fb
7      return fb2
8
9      with open("fib2_input", "r") as f:
10         for s in f:
11             a = [int(x) for x in s.split()]
12
13         t = a[0]
14         if t<0 or t>10**7: print("Ошибка")
15         t2 = LF(t)
16
17         with open("fib2_output", "w") as z:
18             z.write(str(t2))
19             z.close()

```

Объяснение решения: программа открывает файл fib2_input и читает его. С помощью цикла for программа перебирает строку с единственным элементом. Внутри цикла находится переменная со списком, в который попадает единственное число из строки в файле fib2_input. Переменная t-единственный элемент списка a. Если значение переменной t входит в указанный в условии диапазон, то программа не выводит ошибку и продолжает работать с t. С помощью функции LF находится последняя цифра числа Фибоначчи. Затем программа открывает файл fib2_output и записывает строку числа в файл и закрывает его.

Пример работы программы:

Число во входном файле:

```
1 10000000
2
3
```

Число в выходном файле:

```
1 5
2
3
```

Задание 4. Анализ времени работы программ в задании 2 и 3.

Тестирование кода в задании 2

```
1 import time
2 t_start = time.time()
3
4 with open("n4_input_n2", "r") as f:
5     for s in f:
6         a = [int(x) for x in s.split()]
7         n = a[0]
8
9     fb1 = fb2 = 1
10    for i in range(2,n):
11        fb1,fb2 = fb2,fb1+fb2
12    t = str(fb2)
13
14 with open("n4_output_n2", "w") as z:
15     if 0<=n<=45:
16         z.write(str(t))
17         print(time.time() - t_start)
18     else: print("Ошибка")
19     z.close()
```

	Число	Время выполнения
--	-------	------------------

Нижняя граница диапазона значений входных данных из текста задачи	0	0.000134944915771 Process finished
Пример из задачи	10	0.000355958938598 Process finished
Верхняя граница диапазона значений входных данных из текста задачи	45	0.000154018402099 Process finished

Тестирование кода в задании 3

```

1  import time
2  t_start = time.time()
3
4  def LF(x): 2 usages  ± Krimaxev *
5      if x==0: return 1
6      fb1,fb2 = 0,1
7      for i in range(x-1):
8          fb = (fb1+fb2)%10
9          fb1,fb2 = fb2,fb
10         return fb2
11
12  with open("n4_input_n3", "r") as f:
13      for s in f:
14          a = [int(x) for x in s.split()]
15      t = a[0]
16      if t < 0 or t > 10**7: print("Ошибка")
17      t2 = LF(t)
18
19  with open("n4_output_n3", "w") as z:
20      z.write(str(LF(t2)))
21      z.close()
22  print(time.time()-t_start)

```

	Число	Время работы
--	-------	--------------

Нижняя граница диапазона значений входных данных из текста задачи	0	0.00024390220642 Process finished
Пример из задачи	391	0.000284910202026 Process finished
Пример из задачи	327305	0.013142108917236 Process finished
Верхняя граница диапазона значений входных данных из текста задачи	10^{**7}	0.2815580368041992 Process finished wi

Вывод:

В данной лабораторной работе у меня получилось более подробно изучить и отработать процесс работы с файлами. Также я познакомился и поработал с новой библиотекой time, предназначенной для подсчета времени выполнения программы.