**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет инфокоммуникационных технологий

*«Алгоритмы и структуры данных»*

Отчёт к лабораторной работе №1.

**Выполнил**: студент группы К3139,

Ольшанский Григорий Дмитриевич

**Принял**: преподаватель,

Харьковская Татьяна Александровна

Санкт-Петербург

2021 год

**Описание задания:**

Лабораторная работа состоит из четырех задач, для каждой из которых требуется написать реализацию на языке python и описание решения.

**Задание 1. Ввод-вывод**

Необходимо выполнить 4 следующих 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≤ a, b ≤ 10^9. Выход: единственное целое число — результат сложения a + b.

**Реализация:**

a, b = map(int, input().split(‘ ‘))

print(a + b)

**Описание решения:**

Для выполнения этой задачи я воспользовался стандартными функциями языка. Введем 2 числа через пробел, и программа вернет их сумму.

**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 ≤ a, b ≤ 10^9. Выход: единственное целое число — результат сложения a + b^2.

**Реализация:**

a, b = map(int, input().split(‘ ‘))

print(a + b\*b)

**Описание решения:**

Решение аналогично решению задачи 1.

**3. задача a + b с использованием файлов.**

• Имя входного файла: input.txt

• Имя выходного файла: output.txt• Формат входного файла. Входной файл состоит из одной строки, ко- торая содержит два целых числа a и b. Для этих чисел выполняются условия −10^9 ≤ a, b ≤ 10^9

• Формат выходного файла. Выходной файл единственное целое число — результат сложения a + b.

f = open(‘input.txt‘, ‘r‘)

a, b = map(int, f.read().split(‘ ‘))

g = open(‘output.txt‘, ‘w‘)

f.write(a+b)

**Описание решения:**

С помощью функций read и write откроем для чтения файл input.txt, считаем из него 2 числа и запишем их сумму в файл output.txt.

**4. Задача a+b^2 с использованием файлов аналогично предыдущему пункту.**

f = open(‘input.txt‘, ‘r‘)

a, b = map(int, read().split(‘ ‘))

g = open(‘output.txt‘, ‘w‘)

f.write(a+b\*b)

**Описание решения:**

Решение аналогично решению задачи 3.

**Задание 2. Число Фибоначчи.**

Наша цель – разработать эффективный алгоритм для подсчета чисел Фибоначчи.

Определение последовательности Фибоначчи:

F0 = 0 (1)

F1 = 1

Fi = Fi−1 + Fi−2 для i ≥ 2.

• Имя входного файла: input.txt

• Имя выходного файла: output.txt

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

• Формат выходного файла. Число Fn.

Скрипт для отслеживания времени выполнения программы:

import time

start\_time = time.time()

print((time.time() - start\_time))

**Реализация:**

Способ предложенный в задании: рекурсивный

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))

**Итог:**

Функция выполняет расчет числа Фибоначчи за 3.337860107421875e-06 секунд

Способ 1: с помощью цикла.

def fib(n):

a, a1, a2 = 0, 1, 1

while a < n-2:

fib = a1 + a2

a1, a2 = a2, fib

a += 1

return fib

f = open(“input.txt“, “r“)

n = int(f.read())

f = open(“output.txt“, “w“)

f.write(str(fib(n)))

**Описание решения:**

Функция fib вычисляет число Фибоначчи при помощи цикла и тратит на это 2.1457672119140625e-06 секунд

К сожалению, я не силён в алгоритмах, и ничего лучше не придумал, но этот способ явно быстрее.

**Задание 3. Ещё про числа Фибоначчи.**

Определение последней цифры большого числа Фибоначчи.

**Реализация:**

def fib(n):

a, a1, a2 = 0, 1, 1

while a < n-2:

fib = a1 + a2

a1, a2 = a2, fib

a += 1

return fib

f = open(“input.txt“, “r“)

n = int(f.read())

f = open(“output.txt“, “w“)

f.write(str(fib(n)%10))

**Описание решения:**

Алгоритм из задания 3 подходит по критериям памяти и времени. Время выполнения: 3.0994415283203125e-06 секунд и 4.5299530029296875e-06 секунд.

**Вывод:**

В рамках данной лабораторной работы я написал программы для вычисления числа Фибоначчи по его номеру, превосходящие по эффективности заданные в условии работы. Я вспомнил правила написания скриптов, а так же научился подсчитывать время выполнения программы с помощью скрипта на ЯП python.