**Министерство образования и науки Российской Федерации**

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

Факультет программной инженерии и компьютерной техники

Кафедра информатики и прикладной математики   
Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Алгоритмы и структуры данных»

**ОТЧЁТ**

по лабораторной работе №1  
неделя первая

Выполнил:  
Айгузин Иван Олегович   
P3218

Санкт-Петербург

2019 г.

## Задача «a+b»

**Условие задачи**

В данной задаче требуется вычислить сумму двух заданных чисел.

**Формат входного файла**

Входной файл состоит из одной строки, которая содержит два целых числа a и b. Для этих

чисел выполняются условия −10^9 ≤ a ≤ 10^9, −10^9 ≤ b ≤ 10^9.

**Формат выходного файла**

В выходной файл выведите единственное целое число — результат сложения a+b.

**Исходный код**

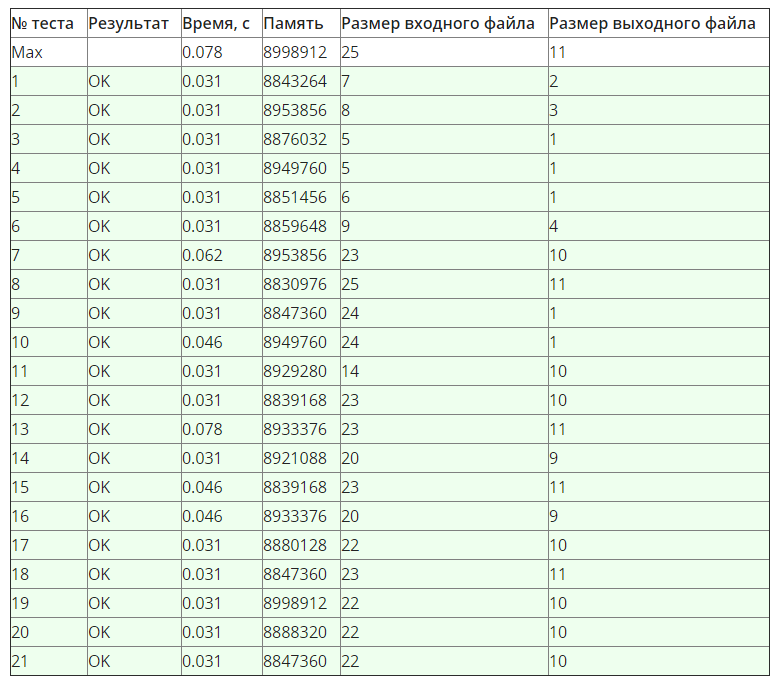
inp = open('input.txt', 'r')

out = open('output.txt', 'w')

a, b = map(int, inp.readline().split())

out.write(str(a + b))

**Результаты**



## Задача «a+b^2»

**Условие задачи**

В данной задаче требуется вычислить значение выражения a + b ^ 2.

**Формат входного файла**

Входной файл состоит из одной строки, которая содержит два целых числа a и b. Для этих

чисел выполняются условия −10^9 ≤ a ≤ 10^9, −10^9 ≤ b ≤ 10^9.

**Формат выходного файла**

В выходной файл выведите единственное целое число — результат выражения a + b ^ 2.

**Исходный код**

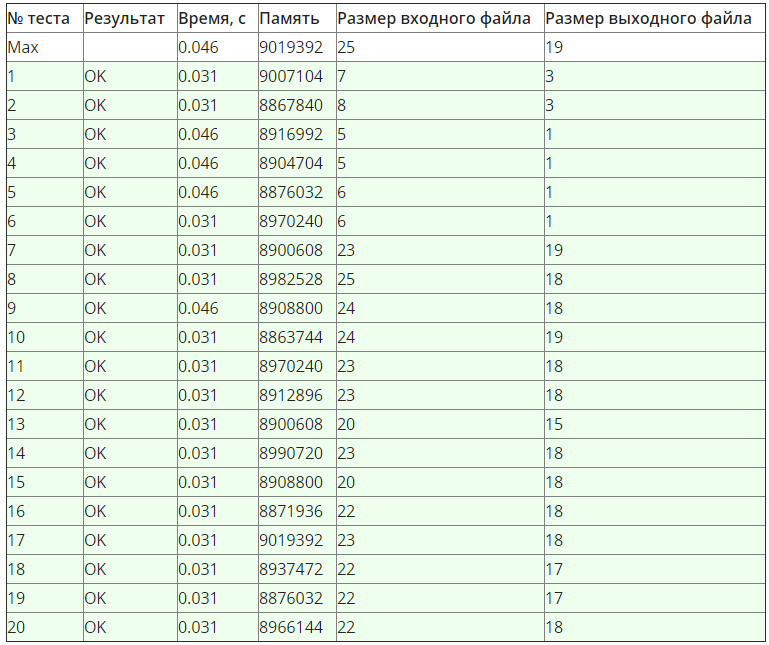
inp = open('input.txt', 'r')

out = open('output.txt', 'w')

a, b = map(int, inp.readline().split())

out.write(str(a + b \*\* 2))

**Результаты**



## Сортировка вставками

**Условие задачи**

Дан массив целых чисел. Ваша задача — отсортировать его в порядке неубывания с помощью сортировки вставками.

Сортировка вставками проходится по всем элементам массива от меньших индексов к большим («слева направо») для каждого элемента определяет его место в предшествующей ему отсортированной части массива и переносит его на это место (возможно, сдвигая некоторые элементы на один индекс вправо). Чтобы проконтролировать, что Вы используете именно сортировку вставками, мы попросим Вас для каждого элемента массива, после того, как он будет обработан, выводить его новый индекс.

**Формат входного файла**

В первой строке входного файла содержится число n (1 <= n <= 1000) — число элементов в массиве. Во второй строке находятся n различных целых чисел, по модулю не превосходящих 10^9

**Формат выходного файла**

В первой строке выходного файла выведите n чисел. При этом i-ое число равно индексу, на который, в момент обработки его сортировкой вставками, был перемещен i-ый элемент исходного массива. Индексы нумеруются, начиная с единицы. Между любыми двумя числами должен стоять ровно один пробел.

Во второй строке выходного файла выведите отсортированный массив. Между любыми двумя числами должен стоять ровно один пробел.

inp = open('input.txt', 'r')

out = open('output.txt', 'w')

n = int(inp.readline())

a = list(map(int, inp.readline().split()))

for i in range(0, n):

j = i

while j > 0 and a[j] < a[j - 1]:

a[j], a[j - 1] = a[j - 1], a[j]

j -= 1

out.write(str(j + 1) + ' ')

out.write('\n')

out.write(' '.join(str(p) for p in a) + '\n')

inp = open('input.txt', 'r')

out = open('output.txt', 'w')

n = int(inp.readline())

a = list(map(float, inp.readline().split()))

b = sorted(((i + 1, a[i]) for i in range(0, n)), key=lambda x: x[1])

out.write(f'{b[0][0]} {b[n // 2][0]} {b[-1][0]}\n')

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

namespace \_5 {

internal class Program {

private static void Main(string[] args) {

using (var input = new StreamReader("./input.txt"))

using (var output = new StreamWriter("./output.txt")) {

Run(input, output);

}

}

private static void Run(StreamReader input, StreamWriter output) {

var n = int.Parse(input.ReadLine());

var numbers = new List<int>(n);

numbers.AddRange(input.ReadLine().Split(' ')

.Select(int.Parse));

for (var i = 0; i < n; i++) {

var min = i;

for (var j = i + 1; j < n; j++) {

if (numbers[min] > numbers[j]) {

min = j;

}

}

if (min == i) {

continue;

}

(numbers[min], numbers[i]) = (numbers[i], numbers[min]);

output.WriteLine($"Swap elements at indices {i + 1} and {min + 1}.");

}

output.WriteLine("No more swaps needed.");

output.WriteLine(string.Join(" ", numbers));

}

}

}