

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
Р3218

Преподаватели:
Романов Алексей Андреевич
Волчек Дмитрий Геннадьевич

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 $-10^9 \leq a \leq 10^9$, $-10^9 \leq b \leq 10^9$.

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

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.078	8998912	25	11
1	OK	0.031	8843264	7	2
2	OK	0.031	8953856	8	3
3	OK	0.031	8876032	5	1
4	OK	0.031	8949760	5	1
5	OK	0.031	8851456	6	1
6	OK	0.031	8859648	9	4
7	OK	0.062	8953856	23	10
8	OK	0.031	8830976	25	11
9	OK	0.031	8847360	24	1
10	OK	0.046	8949760	24	1
11	OK	0.031	8929280	14	10
12	OK	0.031	8839168	23	10
13	OK	0.078	8933376	23	11
14	OK	0.031	8921088	20	9
15	OK	0.046	8839168	23	11
16	OK	0.046	8933376	20	9
17	OK	0.031	8880128	22	10
18	OK	0.031	8847360	23	11
19	OK	0.031	8998912	22	10
20	OK	0.031	8888320	22	10
21	OK	0.031	8847360	22	10

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 $-10^9 \leq a \leq 10^9$, $-10^9 \leq b \leq 10^9$.

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

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.046	9019392	25	19
1	OK	0.031	9007104	7	3
2	OK	0.031	8867840	8	3
3	OK	0.046	8916992	5	1
4	OK	0.046	8904704	5	1
5	OK	0.046	8876032	6	1
6	OK	0.031	8970240	6	1
7	OK	0.031	8900608	23	19
8	OK	0.031	8982528	25	18
9	OK	0.046	8908800	24	18
10	OK	0.031	8863744	24	19
11	OK	0.031	8970240	23	18
12	OK	0.031	8912896	23	18
13	OK	0.031	8900608	20	15
14	OK	0.031	8990720	23	18
15	OK	0.031	8908800	20	18
16	OK	0.031	8871936	22	18
17	OK	0.031	9019392	23	18
18	OK	0.031	8937472	22	17
19	OK	0.031	8876032	22	17
20	OK	0.031	8966144	22	18

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 большему («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 ($1 \leq n \leq 1000$) — 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.

Исходный код

```
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')
```

Результаты

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.406	9297920	10415	14299
1	OK	0.031	9011200	25	43
2	OK	0.031	8929280	7	8
3	OK	0.031	9043968	12	15
4	OK	0.031	8990720	8	11
5	OK	0.046	9039872	10	15
6	OK	0.031	8916992	29	34
7	OK	0.046	8962048	10	15
8	OK	0.031	9019392	10	15
9	OK	0.031	8978432	10	15
10	OK	0.062	8982528	10	15
11	OK	0.031	9035776	10	15
12	OK	0.031	8933376	57	66
13	OK	0.046	8966144	56	65
14	OK	0.031	9019392	57	66
15	OK	0.031	8925184	77	90
16	OK	0.046	9011200	76	89
17	OK	0.031	9072640	77	90
18	OK	0.031	8933376	112	130
19	OK	0.031	9019392	111	130
20	OK	0.031	8957952	110	128
21	OK	0.046	8929280	949	1193
22	OK	0.031	8986624	960	1222
23	OK	0.031	9043968	957	1137
24	OK	0.046	8941568	1490	1891
25	OK	0.046	8970240	1486	1947
26	OK	0.031	9011200	1481	1764
27	OK	0.062	9076736	3723	4891
28	OK	0.031	9097216	3729	5050
29	OK	0.078	9179136	3727	4440
30	OK	0.187	9195520	8456	11341
31	OK	0.046	9191424	8471	11612
32	OK	0.281	9256960	8415	10038
33	OK	0.250	9199616	10415	14038
34	OK	0.031	9232384	10410	14299
35	OK	0.406	9297920	10393	12389

Знакомство с жителями Сортлэнда

1.0 из 1.0 балла (оценивается)

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Владелец графства Сортлэнд, граф Бабблсортер, решил познакомиться со своими подданными. Число жителей в графстве нечетно и составляет n , где n может быть достаточно велико, поэтому граф решил ограничиться знакомством с тремя представителями народонаселения: с самым бедным жителем, с жителем, обладающим средним достатком, и с самым богатым жителем.

Согласно традициям Сортлэнда, считается, что житель обладает средним достатком, если при сортировке жителей по сумме денежных сбережений он оказывается ровно посередине. Известно, что каждый житель графства имеет уникальный идентификационный номер, значение которого расположено в границах от единицы до n . Информация о размере денежных накоплений жителей хранится в массиве M таким образом, что сумма денежных накоплений жителя, обладающего идентификационным номером i , содержится в ячейке $M[i]$. Помогите секретарю графа мистеру Сwoпу вычислить идентификационные номера жителей, которые будут приглашены на встречу с графом.

Формат входного файла

Первая строка входного файла содержит число жителей n ($3 \leq n \leq 9999$, n нечетно). Вторая строка содержит описание массива M , состоящее из n положительных вещественных чисел, разделенных пробелами. Гарантируется, что все элементы массива M различны, а их значения имеют точность не более двух знаков после запятой и не превышают 10^6 .

Формат выходного файла

В выходной файл выведите три целых положительных числа, разделенных пробелами — идентификационные номера беднейшего, среднего и самого богатого жителей Сортлэнда.

```
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')
```

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.062	10493952	98892	16
1	OK	0.046	8986624	30	7
2	OK	0.031	8937472	33	7
3	OK	0.031	9101312	1065	10
4	OK	0.046	8966144	3732	12
5	OK	0.031	9228288	14975	15
6	OK	0.031	9289728	14998	13
7	OK	0.046	9412608	28749	16
8	OK	0.046	9535488	34791	14
9	OK	0.031	9515008	38037	15
10	OK	0.031	9469952	38074	16
11	OK	0.031	9609216	39288	15
12	OK	0.046	9760768	48638	15
13	OK	0.062	9674752	50722	14
14	OK	0.062	9871360	52757	16
15	OK	0.046	9822208	58008	15
16	OK	0.031	9965568	66504	16
17	OK	0.046	10194944	71786	16
18	OK	0.046	10133504	72346	16
19	OK	0.031	10133504	73304	15
20	OK	0.031	10219520	76139	16
21	OK	0.046	10248192	83944	16
22	OK	0.062	10297344	85179	15
23	OK	0.046	10399744	86522	14
24	OK	0.046	10371072	89202	15
25	OK	0.046	10493952	98892	16

Секретарь Своп

Уже знакомый нам из предыдущей задачи граф Бабблсортер поручил своему секретарю, мистеру Свопу, оформлять приглашения беднейшему, богатейшему и среднему по достатку жителю своих владений. Однако кто же, в отсутствие мистера Свopa, будет заниматься самым важным делом — сортировкой массивов чисел? Видимо, это придется сделать Вам!

Дан массив, состоящий из n целых чисел. Вам необходимо его отсортировать по неубыванию. Но делать это нужно так же, как это делает мистер Свop — то есть, каждое действие должно быть взаимной перестановкой пары элементов. Вам также придется записать все, что Вы делали, в файл, чтобы мистер Свop смог проверить Вашу работу.

Формат входного файла

В первой строке входного файла содержится число n ($1 \leq n \leq 5000$) — число элементов в массиве. Во второй строке находятся n целых чисел, по модулю не превосходящих 10^9 . Числа могут совпадать друг с другом.

Формат выходного файла

В первых нескольких строках выведите осуществленные Вами операции перестановки элементов. Каждая строка должна иметь следующий формат:

Swap elements at indices X and Y .

где X и Y — различные индексы массива, элементы на которых нужно переставить ($1 \leq X, Y \leq n$). Мистер Свop любит порядок, поэтому сделайте так, чтобы $X < Y$.

После того, как все нужные перестановки выведены, выведите следующую фразу:

No more swaps needed.

Во последней строке выходного файла выведите отсортированный массив, чтобы мистер Свop не переделывал работу за Вас. Между любыми двумя числами должен стоять ровно один пробел.

```
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));  
}  
}
```

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.078	12976128	51993	255417
1	OK	0.031	11489280	14	139
2	OK	0.031	11395072	7	27
3	OK	0.031	11452416	12	32
4	OK	0.031	11448320	8	63
5	OK	0.031	11440128	10	30
6	OK	0.031	11460608	10	30
7	OK	0.031	11427840	29	49
8	OK	0.031	11468800	10	65
9	OK	0.031	11444224	10	65
10	OK	0.031	11423744	10	100
11	OK	0.031	11497472	10	65
12	OK	0.046	11448320	10	100
13	OK	0.046	11440128	50	140
14	OK	0.031	11501568	56	181
15	OK	0.015	11431936	57	77
16	OK	0.031	11431936	55	145
17	OK	0.031	11472896	75	305
18	OK	0.031	11464704	76	96
19	OK	0.031	11436032	78	203
20	OK	0.031	11493376	108	268
21	OK	0.031	11436032	107	126
22	OK	0.031	11464704	108	303
23	OK	0.046	11534336	948	4177
24	OK	0.031	11460608	947	966
25	OK	0.046	11554816	948	2623
26	OK	0.031	11673600	3720	17384
27	OK	0.046	11567104	3735	3753
28	OK	0.046	11587584	3722	10613
29	OK	0.031	11948032	8463	39804
30	OK	0.031	11665408	8441	8459
31	OK	0.046	11812864	8434	24178
32	OK	0.046	12603392	22822	111246
33	OK	0.031	11948032	22825	22842
34	OK	0.046	12312576	22877	66846
35	OK	0.062	12976128	51987	255417
36	OK	0.078	12587008	51940	51957
37	OK	0.078	12976128	51993	153403