

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1

Неделя первая

Выполнил:

Жумиков Егор Олегович

Преподаватели:

Романов Алексей Андреевич

Волчек Дмитрий Геннадьевич

Оглавление

Задача «a + b».....	3
Условие.....	3
Формат входного файла	3
Формат выходного файла	3
Решение.....	3
Задача «a+b ² »	4
Условие.....	4
Формат входного файла	4
Формат выходного файла	4
Решение.....	4
Задача «Сортировка вставками»	4
Формат входного файла	5
Формат выходного файла	5
Решение.....	5
Задача «Знакомство с жителями Сортлэнда»	7
Условие.....	7
Формат входного файла	7
Формат выходного файла	7
Решение.....	7
Задача «Секретарь Своп».....	9
Условие.....	9
Формат входного файла	9
Формат выходного файла	9
Решение.....	9

Задача «a + b»

Условие

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

Формат выходного файла

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

Решение

```
inp = open('input.txt', 'r')
outp = open('output.txt', 'w')

a, b = map(int, inp.read().split())
outp.write(str(a + b))
outp.close()
```

Результат

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.062	8953856	25	11
1	OK	0.031	8933376	7	2
2	OK	0.062	8826880	8	3
3	OK	0.031	8888320	5	1
4	OK	0.046	8876032	5	1
5	OK	0.031	8818688	6	1
6	OK	0.031	8953856	9	4
7	OK	0.031	8876032	23	10
8	OK	0.046	8843264	25	11
9	OK	0.046	8916992	24	1
10	OK	0.031	8908800	24	1
11	OK	0.031	8851456	14	10
12	OK	0.031	8941568	23	10
13	OK	0.031	8876032	23	11
14	OK	0.031	8953856	20	9
15	OK	0.031	8818688	23	11
16	OK	0.031	8912896	20	9
17	OK	0.031	8855552	22	10
18	OK	0.046	8945664	23	11
19	OK	0.031	8859648	22	10

20	OK	0.031	8949760	22	10
21	OK	0.046	8822784	22	10

Задача «a+b²»

Условие

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 вычисления выражения.

Решение

```
inp = open('input.txt', 'r')
outp = open('output.txt', 'w')

a, b = map(int, inp.read().split())
outp.write(str(a + b**2))
outp.close()
```

Результат

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.062	8998912	25	19
1	OK	0.062	8966144	7	3
2	OK	0.062	8867840	8	3
3	OK	0.031	8916992	5	1
4	OK	0.031	8916992	5	1
5	OK	0.031	8978432	6	1
6	OK	0.046	8904704	6	1
7	OK	0.031	8916992	23	19
8	OK	0.046	8962048	25	18
9	OK	0.031	8847360	24	18
10	OK	0.031	8880128	24	19
11	OK	0.031	8962048	23	18
12	OK	0.031	8843264	23	18
13	OK	0.046	8896512	20	15
14	OK	0.046	8986624	23	18
15	OK	0.031	8843264	20	18

16	OK	0.031	8871936	22	18
17	OK	0.031	8970240	23	18
18	OK	0.031	8880128	22	17
19	OK	0.031	8888320	22	17
20	OK	0.046	8998912	22	18

Задача «Сортировка вставками»

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 ($1 \leq n \leq 1000$) — 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.

Решение

```
from bisect import bisect # модуль бинарного поиска
```

```
inp = open('input.txt', 'r')
outp = open('output.txt', 'w')
```

```
ml = lambda x, y: list(map(x, y))
pr = lambda s: outp.write(s + '\n')
```

```
inp.readline()
arr = ml(int, inp.readline().split())
steps = [0 for i in arr]
res = []
```

```
while len(res) < len(arr):
    # находим место вставки
    steps[len(res)] = bisect(res, arr[len(res)]) + 1
```

```
# вставляем
res.insert(steps[len(res)] - 1, arr[len(res)])
```

```
# выводим
pr(' '.join(ml(str, steps)))
pr(' '.join(ml(str, res)))
outp.close()
```

Результат

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.078	9207808	10415	14298
1	OK	0.031	9027584	25	42
2	OK	0.046	8925184	7	7
3	OK	0.031	9048064	12	14
4	OK	0.078	8929280	8	10
5	OK	0.062	9048064	10	14
6	OK	0.031	9015296	29	33
7	OK	0.031	9039872	10	14
8	OK	0.046	8933376	10	14
9	OK	0.031	8953856	10	14
10	OK	0.031	9109504	10	14
11	OK	0.031	8962048	10	14
12	OK	0.046	8916992	57	65
13	OK	0.031	9064448	56	64
14	OK	0.031	9011200	57	65
15	OK	0.046	8921088	77	89
16	OK	0.031	9043968	76	88
17	OK	0.031	8962048	77	89
18	OK	0.046	8921088	112	129
19	OK	0.031	8990720	111	129
20	OK	0.031	9084928	110	127
21	OK	0.031	8957952	949	1192
22	OK	0.031	9003008	960	1221
23	OK	0.031	9072640	957	1136
24	OK	0.031	8925184	1490	1890
25	OK	0.046	9019392	1486	1946
26	OK	0.046	9076736	1481	1763

27	OK	0.046	8957952	3723	4890
28	OK	0.031	9064448	3729	5049
29	OK	0.078	9109504	3727	4439
30	OK	0.031	9084928	8456	11340
31	OK	0.031	9093120	8471	11611
32	OK	0.031	9162752	8415	10037
33	OK	0.046	9175040	10415	14037
34	OK	0.046	9207808	10410	14298
35	OK	0.046	9199616	10393	12388

Задача «Знакомство с жителями Сортлэнда»

Условие

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 Свопу 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.

Решение

```
from bisect import bisect # binary search
```

```
inp = open('input.txt', 'r')
outp = open('output.txt', 'w')
```

```
ml = lambda x, y: list(map(x, y))
pr = lambda s: outp.write(s + '\n')
```

```

inp.readline() # read n
arr = ml(float, inp.readline().split())
# convert list of wealths to list with indecies
arr = list(((i + 1), v) for i, v in enumerate(arr))

# sort it by wealth
arr = sorted(arr, key=lambda a: a[1])

# print output
pr(f'{arr[0][0]} {arr[len(arr) // 2][0]} {arr[-1][0]}')

outp.close()

```

Результат

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.109	10932224	98892	16
1	OK	0.031	9023488	30	7
2	OK	0.031	9060352	33	7
3	OK	0.031	9019392	1065	10
4	OK	0.031	9035776	3732	12
5	OK	0.046	9216000	14975	15
6	OK	0.031	9326592	14998	13
7	OK	0.046	9515008	28749	16
8	OK	0.031	9572352	34791	14
9	OK	0.046	9756672	38037	15
10	OK	0.031	9687040	38074	16
11	OK	0.046	9736192	39288	15
12	OK	0.062	9859072	48638	15
13	OK	0.031	9891840	50722	14
14	OK	0.046	10022912	52757	16
15	OK	0.046	9928704	58008	15
16	OK	0.078	10199040	66504	16
17	OK	0.078	10330112	71786	16
18	OK	0.078	10207232	72346	16
19	OK	0.062	10305536	73304	15
20	OK	0.062	10448896	76139	16
21	OK	0.046	10559488	83944	16
22	OK	0.109	10645504	85179	15
23	OK	0.078	10801152	86522	14

24	OK	0.046	10604544	89202	15
25	OK	0.078	10932224	98892	16

Задача «Секретарь Своп»

Условие

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 Свопа, 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 Своп — то есть, каждое действие должно быть взаимной перестановкой пары элементов. Вам также придется записать все, что Вы делали, в файл, чтобы мистер Своп 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 109. 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 Своп 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 Своп не переделывал работу за Вас. Между любыми двумя числами должен стоять ровно один пробел.

Решение

```
using System;
using System.IO;
using System.Linq;
```

```
namespace Temp
{
    public static class Program
    {
        private static StreamReader sr;
        private static StreamWriter sw;
        private static int[] a;

        private static int Partition(int f, int l)
        {
            int pivotValue = a[f];
            int leftMark = f + 1, rightMark = l;
```

```

        while (rightMark ≥ leftMark)
        {
            while (leftMark ≤ rightMark && a[leftMark] ≤
pivotValue)
            {
                leftMark++;
            }

            while (rightMark ≥ leftMark && a[rightMark] ≥
pivotValue)
            {
                rightMark--;
            }

            if (rightMark ≥ leftMark)
            {
                Swap(leftMark, rightMark);
            }
        }

        Swap(f, rightMark);

        return rightMark;
    }

    private static void Swap(int left, int right)
    {
        if (left == right) return;

        if (left > right)
        {
            int tmp = left;
            left = right;
            right = tmp;
        }

        sw.WriteLine($"Swap elements at indices {left + 1} and
{right + 1}.");

        int tmp = a[left];
        a[left] = a[right];
        a[right] = tmp;
    }

    private static void Qsort(int f, int l)
    {
        while (f < l)
        {
            int splitPoint = Partition(f, l);

```

```

        if (splitPoint - f < l - splitPoint)
        {
            Qsort(f, splitPoint - 1);
            f = splitPoint + 1;
        }
        else
        {
            Qsort(splitPoint + 1, l);
            l = splitPoint - 1;
        }
    }

}

static void Main(string[] args)
{
    sr = new StreamReader("input.txt");
    sw = new StreamWriter("output.txt", false);

    int n = int.Parse(sr.ReadLine());
    a = sr.ReadLine()
        .Split()
        .Select(s => int.Parse(s))
        .ToArray();
    sr.Close();

    Qsort(0, a.Length - 1);
    sw.WriteLine("No more swaps needed.");
    sw.WriteLine(string.Join(" ", a));

    sw.Close();
}
}
}

```

Результат

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.062	12873728	51993	602899
1	OK	0.031	11358208	14	174
2	OK	0.031	11247616	7	27
3	OK	0.046	11300864	12	32
4	OK	0.031	11333632	8	63
5	OK	0.031	11329536	10	65
6	OK	0.031	11386880	10	30
7	OK	0.031	11350016	29	49
8	OK	0.031	11317248	10	65

9	OK	0.046	11341824	10	65
10	OK	0.031	11313152	10	100
11	OK	0.046	11362304	10	65
12	OK	0.031	11292672	10	100
13	OK	0.046	11317248	50	140
14	OK	0.031	11370496	56	181
15	OK	0.015	11354112	57	77
16	OK	0.031	11300864	55	145
17	OK	0.015	11350016	75	305
18	OK	0.031	11300864	76	96
19	OK	0.062	11333632	78	203
20	OK	0.031	11345920	108	268
21	OK	0.031	11304960	107	126
22	OK	0.031	11345920	108	303
23	OK	0.031	11419648	948	5630
24	OK	0.046	11370496	947	966
25	OK	0.031	11378688	948	2623
26	OK	0.015	11706368	3720	29762
27	OK	0.031	11350016	3735	3753
28	OK	0.031	11509760	3722	10613
29	OK	0.031	12103680	8463	75738
30	OK	0.031	11583488	8441	8459
31	OK	0.031	11677696	8434	24178
32	OK	0.031	12554240	22822	243536
33	OK	0.046	11833344	22825	22842
34	OK	0.046	12210176	22877	66846
35	OK	0.046	12873728	51987	602899
36	OK	0.062	12472320	51940	51957
37	OK	0.046	12795904	51993	153403