Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа № 1

по дисциплине «Алгоритмы и структуры данных»

Openedu – неделя 1

Подготовил:

студент группы P3217

Бураков Илья Алексеевич

Преподаватели:

Романов Алексей Андреевич

Волчек Дмитрий Геннадьевич

Санкт-Петербург, 2019

Задача «a+b»

Условие

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

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

Формат выходного файла

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

Примеры

input.txt	output.txt
23 11	34
-100 1	-99

Решение

openedu/week1/lab1_1.py

```
from edx_io import edx_io

# testing official course io wrapper for Python
with edx_io() as io:
    io.writeln(io.next_int() + io.next_int())
```

Результаты

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.078	11091968	25	12
1	OK	0.046	11022336	7	3
2	OK	0.046	11091968	8	4
3	OK	0.046	10977280	5	2
4	OK	0.062	10993664	5	2
5	OK	0.078	10903552	6	2
6	OK	0.062	10969088	9	5
7	OK	0.062	11022336	23	11
8	OK	0.062	10977280	25	12

9	OK	0.062	10960896	24	2
10	OK	0.078	11005952	24	2
11	OK	0.062	11018240	14	11
12	OK	0.078	10969088	23	11
13	OK	0.062	10883072	23	12
14	OK	0.078	11030528	20	10
15	OK	0.046	11005952	23	12
16	OK	0.062	11005952	20	10
17	OK	0.046	11059200	22	11
18	OK	0.078	10964992	23	12
19	OK	0.062	11042816	22	11
20	OK	0.062	11034624	22	11
21	OK	0.062	10985472	22	11

Задача «a+b^2»

Условие

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

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

Формат выходного файла

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

Примеры

input.txt	output.txt
23 11	144
-100 1	-99

Решение

openedu/week1/lab1_2.py

```
from edx_io import edx_io

with edx_io() as io:
    io.writeln(io.next_int() + io.next_int() ** 2)
```

Результаты

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.078	11083776	25	20
1	OK	0.046	11063296	7	4
2	OK	0.046	10899456	8	4
3	OK	0.078	11010048	5	2
4	OK	0.062	10989568	5	2
5	OK	0.062	11034624	6	2
6	OK	0.062	10997760	6	2
7	OK	0.062	11014144	23	20
8	OK	0.078	10969088	25	19
9	OK	0.078	10981376	24	19
10	OK	0.046	11026432	24	20
11	OK	0.046	11034624	23	19
12	OK	0.062	11034624	23	19
13	OK	0.046	11022336	20	16
14	OK	0.062	10944512	23	19
15	OK	0.046	11083776	20	19
16	OK	0.062	10997760	22	19
17	OK	0.062	10969088	23	19
18	OK	0.062	11010048	22	18
19	OK	0.062	11022336	22	18
20	OK	0.062	10956800	22	19

Сортировка вставками

Условие

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

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 ($1 \le n \le 1000$) — 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.

Пример

input.txt	output.txt
10 1 8 4 2 3 7 5 6 9 0	1 2 2 2 3 5 5 6 9 1 0 1 2 3 4 5 6 7 8 9

Решение

openedu/week1/lab1_3.py

```
from edx_io import edx_io

with edx_io() as io:
    # read input
    n = io.next_int()
    arr = [io.next_int() for i in range(n)]

    # list for saving intermediate indices
    indices = []

    # perform soring
    for i in range(n):
        for j in range(i - 1, -1, -1):
            if arr[j] > arr[j+1]:
                # if the order is wrong, swap 'em all!
                arr[j], arr[j+1] = arr[j+1], arr[j]
            else:
                # else remember the index or the inserted element, as requested
                # need j+1, indexing from 1
                indices.append(j + 2)
                # and go for inserting the next element
                break
        else:
            # if we reached the beginning, put 1 as index
            indices.append(1)

    # write results
    io.writeln(indices)
    io.writeln(arr)
```

Результаты

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.375	11247616	10415	14296
1	OK	0.046	11083776	25	40
2	OK	0.046	11014144	7	5
3	OK	0.062	10948608	12	12
4	OK	0.062	10989568	8	8

5	OK	0.046	11010048	10	12
6	OK	0.046	10989568	29	31
7	OK	0.062	10997760	10	12
8	OK	0.078	11001856	10	12
9	OK	0.062	11014144	10	12
10	OK	0.093	11026432	10	12
11	OK	0.046	11001856	10	12
12	OK	0.046	11005952	57	63
13	OK	0.062	10956800	56	62
14	OK	0.062	10928128	57	63
15	OK	0.078	11059200	77	87
16	OK	0.078	10993664	76	86
17	OK	0.046	11034624	77	87
18	OK	0.062	11014144	112	127
19	OK	0.046	10985472	111	127
20	OK	0.046	11030528	110	125
21	OK	0.046	11005952	949	1190
22	OK	0.062	11022336	960	1219
23	OK	0.046	11018240	957	1134
24	OK	0.062	11051008	1490	1888
25	OK	0.062	11005952	1486	1944
26	OK	0.078	11067392	1481	1761
27	OK	0.078	10964992	3723	4888
28	OK	0.062	11026432	3729	5047
29	OK	0.093	11067392	3727	4437
30	OK	0.203	11223040	8456	11338
31	OK	0.062	11247616	8471	11609
32	OK	0.250	11202560	8415	10035
33	OK	0.281	11227136	10415	14035
34	OK	0.062	11243520	10410	14296
35	OK	0.375	11218944	10393	12386

Знакомство с жителями Сортлэнда

Условие

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 Свопу вычислить идентификационные номера жителей, которые будут приглашены на встречу с графом.

Формат входного файла

Первая строка входного файла содержит число жителей n ($3 \le n \le 9999$, 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.

Пример

input.txt	output.txt
5 10.00 8.70 0.01 5.00 3.00	3 4 1

Решение

openedu/week1/lab1_4.py

```
from edx_io import edx_io

with edx_io() as io:
    # read input
    n = io.next_int()
    incomes = [(io.next_float(), i + 1) for i in range(n)]

    # perform sorting
    # I tried Insertion Sort, even with Cython optimizations - time limit exceeded
:(
    # bad time or variable limits, imo
    # thus using stable O(nlogn) sort here, they deserve it
    incomes.sort()

    # write results
    io.writeln((incomes[0][1], incomes[n // 2][1], incomes[-1][1]))
```

Результаты

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.093	13074432	98892	15
1	OK	0.078	11026432	30	6
2	OK	0.093	11005952	33	6

3	OK	0.062	11042816	1065	9
4	OK	0.078	11030528	3732	11
5	OK	0.062	11247616	14975	14
6	OK	0.062	11313152	14998	12
7	OK	0.062	11636736	28749	15
8	OK	0.062	11730944	34791	13
9	OK	0.062	11771904	38037	14
10	OK	0.062	11792384	38074	15
11	OK	0.093	11841536	39288	14
12	OK	0.078	11931648	48638	14
13	OK	0.062	12107776	50722	13
14	OK	0.062	12111872	52757	15
15	OK	0.062	12255232	58008	14
16	OK	0.078	12378112	66504	15
17	OK	0.078	12505088	71786	15
18	OK	0.062	12460032	72346	15
19	OK	0.078	12587008	73304	14
20	OK	0.093	12587008	76139	15
21	OK	0.078	12808192	83944	15
22	OK	0.078	12869632	85179	14
23	OK	0.093	12877824	86522	13
24	OK	0.078	12881920	89202	14
25	OK	0.078	13074432	98892	15

Секретарь Своп

Условие

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

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 ($1 \le n \le 5000$) — 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 ($1 \le X, Y \le n$). 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.

Пример

input.txt	output.txt
5 3 1 4 2 2	Swap elements at indices 1 and 2. Swap elements at indices 2 and 4. Swap elements at indices 3 and 5. No more swaps needed. 1 2 2 3 4

Решение

openedu/week1/lab1_5.py

```
from edx_io import edx_io

with edx_io() as io:
    # read input
    n = io.next_int()
    incomes = [io.next_int() for i in range(n)]

    # perform sorting swaps not to bloat io (selection sort)
    for i in range(n - 1):
        # find min value and its index
        mi = incomes.index(min(incomes[i:]), i)

        # swap current and min
        if mi != i:
            incomes[mi], incomes[i] = incomes[i], incomes[mi]
            io.writeln(f"Swap elements at indices {i + 1} and {mi + 1}.")

    # write results
    io.writeln("No more swaps needed.")
    io.writeln(incomes)
```

Результаты

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.718	12808192	51993	250424
1	OK	0.078	11014144	14	134
2	OK	0.046	11046912	7	25
3	OK	0.078	11010048	12	30
4	OK	0.078	10997760	8	60
5	OK	0.093	11042816	10	28
6	OK	0.062	11091968	10	28
7	OK	0.078	10928128	29	47
8	OK	0.062	11030528	10	62
9	OK	0.062	11014144	10	62
10	OK	0.062	11042816	10	96
11	OK	0.046	11026432	10	62
12	OK	0.093	10985472	10	96
13	OK	0.046	10989568	50	136
14	OK	0.062	11063296	56	176
15	OK	0.093	11051008	57	75
16	OK	0.062	11038720	55	141
17	OK	0.062	11034624	75	297
18	OK	0.046	10940416	76	94
19	OK	0.062	11038720	78	198
20	OK	0.046	11042816	108	262
21	OK	0.062	11022336	107	124
22	OK	0.062	11055104	108	296
23	OK	0.046	10985472	948	4088
24	OK	0.046	11038720	947	964
25	OK	0.062	11001856	948	2576
26	OK	0.062	11198464	3720	17029
27	OK	0.062	11141120	3735	3751
28	OK	0.062	11157504	3722	10432
29	OK	0.078	11386880	8463	38996
30	OK	0.062	11214848	8441	8457
31	OK	0.093	11264000	8434	23770
32	OK	0.203	11796480	22822	109054
33	OK	0.156	11689984	22825	22840
34	OK	0.187	11743232	22877	65745
35	OK	0.718	12808192	51987	250424
36	OK	0.500	12382208	51940	51955
37	OK	0.671	12492800	51993	150901