

Санкт-Петербургский Научный Исследовательский Университет
Информационных Технологий, Механики и Оптики

Задачи первой недели
по курсу «Алгоритмы и структуры данных»
на Openedu

Выполнил: студент группы Р3218
Артамонов Александр Владимирович

Санкт-Петербург, 2019г

Задача 1. «a + b»

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 $a + b$.

Примеры

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

Код программы (C++)

```
#include <fstream>
using namespace std;

int main() {
    long a, b;

    ifstream input("input.txt");
    input >> a >> b;
    input.close();

    long long sum = a + b;

    ofstream output("output.txt");
    output << sum;
    output.close();

    return 0;
}
```

Бенчмарк (задача 1)

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.031	2383872	25	11
1	OK	0.000	2367488	7	2
2	OK	0.000	2371584	8	3
3	OK	0.000	2367488	5	1
4	OK	0.015	2367488	5	1
5	OK	0.015	2371584	6	1
6	OK	0.000	2371584	9	4
7	OK	0.000	2371584	23	10
8	OK	0.015	2383872	25	11
9	OK	0.000	2371584	24	1
10	OK	0.000	2371584	24	1
11	OK	0.015	2367488	14	10
12	OK	0.000	2371584	23	10
13	OK	0.015	2371584	23	11
14	OK	0.015	2371584	20	9
15	OK	0.000	2371584	23	11
16	OK	0.000	2371584	20	9
17	OK	0.031	2371584	22	10
18	OK	0.015	2371584	23	11
19	OK	0.000	2371584	22	10
20	OK	0.000	2371584	22	10
21	OK	0.000	2367488	22	10

Задача 2. «a + b²»

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 $a+b^2$.

Примеры

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

Код программы (C++)

```
#include <fstream>
using namespace std;

int main() {
    long long a, b;

    ifstream input("input.txt");
    input >> a >> b;
    input.close();

    long long sum = a + b * b;

    ofstream output("output.txt");
    output << sum;
    output.close();

    return 0;
}
```

Бенчмарк (задача 2)

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.031	2383872	25	19
1	OK	0.000	2371584	7	3
2	OK	0.015	2371584	8	3
3	OK	0.015	2371584	5	1
4	OK	0.000	2371584	5	1
5	OK	0.015	2371584	6	1
6	OK	0.000	2371584	6	1
7	OK	0.000	2367488	23	19
8	OK	0.000	2371584	25	18
9	OK	0.015	2367488	24	18
10	OK	0.000	2371584	24	19
11	OK	0.000	2383872	23	18
12	OK	0.031	2383872	23	18
13	OK	0.015	2371584	20	15
14	OK	0.000	2371584	23	18
15	OK	0.015	2383872	20	18
16	OK	0.015	2371584	22	18
17	OK	0.000	2367488	23	18
18	OK	0.015	2383872	22	17
19	OK	0.015	2371584	22	17
20	OK	0.000	2367488	22	18

Задача 3. 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.

Пример

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

Код программы (C++)

```
#include <fstream>
using namespace std;

int main() {

    int n;

    //Ввод длины и самого массива
    ifstream input("input.txt");
    input >> n;
    long* a = new long[n];
    for (int i = 0; i < n; i++) {
        input >> a[i];
    }
    input.close();

    ofstream output("output.txt");

    //Первый элемент всегда остаётся на месте
    output << 1 << " ";

    int j;
    long temp;

    //Алгоритм сортировки вставками
    for (int i = 1; i < n; i++) {
        j = i;
        while (j != 0 && a[j] < a[j - 1]) {
            temp = a[j];
            a[j] = a[j - 1];
            a[j - 1] = temp;
            j--;
        }
        //В массиве индекс с 0, а в ответе должны быть с 1
        output << j + 1;
        //После каждого числа кроме последнего должен стоять пробел
        if (i != n - 1) output << " ";
    }
    //Отсортированный массив должен выводиться с новой строки
    output << endl;

    //Вывод массива
    for (int i = 0; i < n; i++) {
        output << a[i];
        if (i != n - 1) output << " ";
    }

    output.close();

    return 0;
}
```

Бенчмарк (задача 3)

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.031	2592768	10415	14296
1	OK	0.000	2592768	25	40
2	OK	0.015	2576384	7	6
3	OK	0.015	2576384	12	12
4	OK	0.000	2355200	8	8
5	OK	0.000	2367488	10	12
6	OK	0.031	2367488	29	31
7	OK	0.000	2367488	10	12
8	OK	0.000	2367488	10	12
9	OK	0.000	2355200	10	12
10	OK	0.000	2367488	10	12
11	OK	0.015	2355200	10	12
12	OK	0.000	2367488	57	63
13	OK	0.000	2355200	56	62
14	OK	0.000	2355200	57	63
15	OK	0.000	2355200	77	87
16	OK	0.000	2367488	76	86
17	OK	0.000	2351104	77	87
18	OK	0.015	2355200	112	127
19	OK	0.015	2367488	111	127
20	OK	0.015	2355200	110	125
21	OK	0.000	2355200	949	1190
22	OK	0.000	2355200	960	1219
23	OK	0.000	2367488	957	1134
24	OK	0.000	2355200	1490	1888
25	OK	0.015	2367488	1486	1944
26	OK	0.015	2351104	1481	1761
27	OK	0.015	2355200	3723	4888
28	OK	0.015	2375680	3729	5047
29	OK	0.000	2355200	3727	4437
30	OK	0.000	2359296	8456	11338
31	OK	0.000	2359296	8471	11609
32	OK	0.000	2371584	8415	10035
33	OK	0.000	2371584	10415	14035
34	OK	0.000	2355200	10410	14296
35	OK	0.000	2359296	10393	12386

Задача 4. Знакомство с жителями Сортлэнда

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.

Пример

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

Код программы (C++)

```
#include <fstream>
using namespace std;

int main() {

    int n;

    ifstream input("input.txt");
    input >> n;

    //Инициализация двумерного массива, каждый элемент которого хранит состояние
    жителя в [x][1] и его идентификатор в [x][0]
    float** M = new float*[n];
    for (int i = 0; i < n; i++) {
        M[i] = new float[2];
        M[i][0] = i + 1;
        input >> M[i][1];
    }

    input.close();

    int j;
    float* temp;

    //Сортировка вставками
    for (int i = 1; i < n; i++) {
        j = i;
        while (j != 0 && M[j][1] < M[j - 1][1]) {
            temp = M[j];
            M[j] = M[j - 1];
            M[j - 1] = temp;
            j--;
        }
    }

    int middlePos = n / 2;
    int min = M[0][0];
    int middle = M[middlePos][0];
    int max = M[n - 1][0];

    ofstream output("output.txt");
    output << min << " " << middle << " " << max;
    output.close();

    return 0;
}
```

Бенчмарк (задача 4)

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.125	2637824	98892	14
1	OK	0.031	2379776	30	5
2	OK	0.015	2379776	33	5
3	OK	0.015	2383872	1065	8
4	OK	0.000	2400256	3732	10
5	OK	0.015	2424832	14975	13
6	OK	0.000	2420736	14998	11
7	OK	0.015	2478080	28749	14
8	OK	0.015	2486272	34791	12
9	OK	0.015	2494464	38037	13
10	OK	0.031	2490368	38074	14
11	OK	0.031	2490368	39288	13
12	OK	0.015	2531328	48638	13
13	OK	0.031	2527232	50722	12
14	OK	0.031	2535424	52757	14
15	OK	0.046	2539520	58008	13
16	OK	0.062	2600960	66504	14
17	OK	0.062	2580480	71786	14
18	OK	0.046	2580480	72346	14
19	OK	0.046	2592768	73304	13
20	OK	0.062	2584576	76139	14
21	OK	0.078	2637824	83944	14
22	OK	0.078	2625536	85179	13
23	OK	0.078	2629632	86522	12
24	OK	0.078	2629632	89202	13
25	OK	0.125	2633728	98892	14

Задача 5. 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 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 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 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	Swap elements at indices 1 and 2.
3 1 4 2 2	Swap elements at indices 2 and 4.
	Swap elements at indices 3 and 5.
	No more swaps needed.
	1 2 2 3 4

Код программы (C++)

```
#include <fstream>
using namespace std;

int main() {

    int n;

    ifstream input("input.txt");

    input >> n;
    long* M = new long[n];

    for (int i = 0; i < n; i++) {
        input >> M[i];
    }

    input.close();

    int j;
    long temp;
    int min;

    ofstream output("output.txt");

    for (int i = 0; i < n; i++) {
        j = i + 1;
        min = i;
        temp = M[i];
        //Для каждого i-того элемента осуществляется поиск наименьшего элемента
        //справа от него
        while (j < n) {
            if (temp > M[j]) {
                temp = M[j];
                min = j;
            }
            j++;
        }
        //Если этот элемент найден, то он меняется с i-тым, получается, что
        //элементы по возрастанию становятся с начала
        if (i != min) {
            M[min] = M[i];
            M[i] = temp;
            output << "Swap elements at indices " << i + 1 << " and " << min + 1
            << "." << endl;
        }
        output << "No more swaps needed." << endl;

        for (int i = 0; i < n; i++) {
            output << M[i];
            if (i != n - 1) output << " ";
        }

        output.close();

        return 0;
    }
```

Бенчмарк (задача 5)

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.062	2392064	51993	255415
1	OK	0.015	2359296	14	137
2	OK	0.015	2355200	7	25
3	OK	0.015	2359296	12	30
4	OK	0.000	2359296	8	61
5	OK	0.000	2359296	10	28
6	OK	0.000	2359296	10	28
7	OK	0.015	2351104	29	47
8	OK	0.000	2355200	10	63
9	OK	0.000	2359296	10	63
10	OK	0.031	2359296	10	98
11	OK	0.000	2371584	10	63
12	OK	0.015	2355200	10	98
13	OK	0.000	2371584	50	138
14	OK	0.000	2359296	56	179
15	OK	0.000	2359296	57	75
16	OK	0.000	2359296	55	143
17	OK	0.000	2359296	75	303
18	OK	0.000	2359296	76	94
19	OK	0.015	2359296	78	201
20	OK	0.015	2359296	108	266
21	OK	0.015	2371584	107	124
22	OK	0.000	2359296	108	301
23	OK	0.015	2355200	948	4175
24	OK	0.000	2359296	947	964
25	OK	0.000	2359296	948	2621
26	OK	0.015	2363392	3720	17382
27	OK	0.000	2363392	3735	3751
28	OK	0.015	2375680	3722	10611
29	OK	0.015	2363392	8463	39802
30	OK	0.015	2363392	8441	8457
31	OK	0.000	2363392	8434	24176
32	OK	0.015	2371584	22822	111244
33	OK	0.031	2371584	22825	22840
34	OK	0.015	2371584	22877	66844
35	OK	0.062	2387968	51987	255415
36	OK	0.031	2392064	51940	51955
37	OK	0.046	2387968	51993	153401