

Министерство образования и науки Российской Федерации
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ

Факультет программной инженерии и компьютерной техники
Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Алгоритмы и структуры данных»

ОТЧЁТ

по лабораторной работе №1 (Week 1 Openedu)

Студент Дунаев Алексей Игоревич

группа Р3217

Преподаватель Муромцев Дмитрий Ильич

Санкт-Петербург

2019 г.

Содержание

Задача 1 « $a+b$ ».....	3
Исходный код к задаче 1.....	3
Бенчмарк к задаче 1.....	4
Задача 2 « $a+b^2$ ».....	4
Исходный код к задаче 2.....	5
Бенчмарк к задаче 2.....	5
Задача 3 Сортировка вставками.....	6
Исходный код к задаче 3.....	7
Бенчмарк к задаче 3.....	8

Задача 1 «a+b»

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

Формат выходного файла

В выходной файл выведите единственное целое число — результат сложения.

Примеры

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

Исходный код к задаче 1

```
#include <iostream>
#include <fstream>
using namespace std;
int main () {
    ifstream fi;
    ofstream fo;
    int a,b;
    fi.open ("input.txt");
    fo.open ("output.txt");
    fi >> a >> b;
    fo << a+b;
    fi.close();
    fo.close();
    return 0;
}
```

Бенчмарк к задаче 1

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.046	3612672	25	11
1	OK	0.046	3571712	7	2
2	OK	0.015	3584000	8	3
3	OK	0.015	3600384	5	1
4	OK	0.015	3579904	5	1
5	OK	0.015	3600384	6	1
6	OK	0.015	3579904	9	4
7	OK	0.000	3592192	23	10
8	OK	0.015	3579904	25	11
9	OK	0.015	3596288	24	1
10	OK	0.000	3567616	24	1
11	OK	0.015	3588096	14	10
12	OK	0.000	3592192	23	10
13	OK	0.031	3588096	23	11
14	OK	0.000	3575808	20	9
15	OK	0.000	3575808	23	11
16	OK	0.000	3579904	20	9
17	OK	0.015	3571712	22	10
18	OK	0.000	3604480	23	11
19	OK	0.000	3604480	22	10
20	OK	0.015	3608576	22	10
21	OK	0.015	3612672	22	10

Задача 2 « $a+b^2$ »

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

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

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

Исходный код к задаче 2

```
#include <iostream>
#include <fstream>
#include <inttypes.h>
using namespace std;
int main () {
    ifstream fi;
    ofstream fo;
    int64_t a,b;
    fi.open ("input.txt");
    fo.open ("output.txt");
    fi >> a >> b;
    fo << a+b*b;
    fi.close();
    fo.close();
    return 0;
}
```

Бенчмарк к задаче 2

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.031	3641344	25	19
1	OK	0.015	3608576	7	3
2	OK	0.000	3616768	8	3
3	OK	0.015	3608576	5	1
4	OK	0.000	3596288	5	1
5	OK	0.000	3616768	6	1
6	OK	0.000	3612672	6	1
7	OK	0.000	3600384	23	19
8	OK	0.015	3596288	25	18
9	OK	0.015	3624960	24	18
10	OK	0.015	3592192	24	19
11	OK	0.000	3604480	23	18
12	OK	0.000	3596288	23	18
13	OK	0.000	3620864	20	15
14	OK	0.000	3600384	23	18
15	OK	0.031	3612672	20	18
16	OK	0.000	3600384	22	18
17	OK	0.000	3641344	23	18
18	OK	0.015	3612672	22	17
19	OK	0.015	3624960	22	17
20	OK	0.015	3596288	22	18

Задача 3 Сортировка вставками

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 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 различные 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
1 8 4 2 3 7 5 6	9 1
9 0	0 1 2 3 4 5 6 7
	8 9

Комментарий к примеру

В примере сортировка вставками работает следующим образом:

1. Первый элемент остается на своем месте, поэтому первое число в ответе — единица. Отсортированная часть массива: [1]
2. Второй элемент больше первого, поэтому он тоже остается на своем месте, и второе число в ответе — двойка. [1 8]
3. Четверка меньше восьмерки, поэтому занимает второе место. [1 4 8]
4. Двойка занимает второе место. [1 2 4 8]
5. Тройка занимает третье место. [1 2 3 4 8]
6. Семерка занимает пятое место. [1 2 3 4 7 8]
7. Пятерка занимает пятое место. [1 2 3 4 5 7 8]
8. Шестерка занимает шестое место. [1 2 3 4 5 6 7 8]
9. Девятка занимает девятое место. [1 2 3 4 5 6 7 8 9]
10. Ноль занимает первое место. [0 1 2 3 4 5 6 7 8 9]

Исходный код к задаче 3

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <fstream>
#include <inttypes.h>
using namespace std;
int main () {
    ifstream fi;
    ofstream fo;
    size_t n;
    int tmp;
    vector<int> v;
    fi.open ("input.txt");
    fo.open ("output.txt");
    fi >> n;
    for (size_t i = 0; i < n; ++i) {
        fi >> tmp;
        v.push_back(tmp);
    }
    fo << 1 << " ";
    for (size_t i = 1; i < n; ++i) {
        int key = v[i];
        int j = i-1;
        while (j >= 0 and v[j] > key) {
            v[j+1] = v[j];
            j--;
        }
        v[j+1] = key;
        fo << j + 2 << " "; // TODO magic consts
    }
    fo << "\n";
    for (size_t i = 0; i < n; ++i) {
        fo << v[i] << " ";
    }
    fi.close();
    fo.close();
    return 0;
}
```

Бенчмарк к задаче 3

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.046	3665920	10415	14298
1	OK	0.015	3637248	25	42
2	OK	0.015	3661824	7	7
3	OK	0.000	3629056	12	14
4	OK	0.000	3641344	8	10
5	OK	0.000	3633152	10	14
6	OK	0.015	3633152	29	33

7	OK	0.015	3616768	10	14
8	OK	0.046	3641344	10	14
9	OK	0.015	3637248	10	14
10	OK	0.000	3645440	10	14
11	OK	0.015	3624960	10	14
12	OK	0.000	3653632	57	65
13	OK	0.000	3624960	56	64
14	OK	0.046	3641344	57	65
15	OK	0.000	3641344	77	89
16	OK	0.031	3629056	76	88
17	OK	0.000	3637248	77	89
18	OK	0.000	3624960	112	129
19	OK	0.000	3624960	111	129
20	OK	0.015	3624960	110	127
21	OK	0.015	3637248	949	1192
22	OK	0.000	3620864	960	1221
23	OK	0.000	3624960	957	1136
24	OK	0.046	3633152	1490	1890
25	OK	0.000	3649536	1486	1946
26	OK	0.015	3637248	1481	1763
27	OK	0.015	3637248	3723	4890
28	OK	0.000	3641344	3729	5049
29	OK	0.015	3665920	3727	4439
30	OK	0.000	3633152	8456	11340
31	OK	0.000	3620864	8471	11611
32	OK	0.015	3649536	8415	10037
33	OK	0.015	3633152	10415	14037
34	OK	0.000	3620864	10410	14298
35	OK	0.000	3649536	10393	12388

Задача 4 Знакомство с жителями Сортлэнда

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 (нечетно). Вторая строка содержит описание массива M , состоящее из 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

Комментарий к примеру

Если отсортировать жителей по их достатку, получится следующий массив:

[0.01, 3] [3.00, 5] [5.00, 4] [8.70, 2] [10.00, 1]

Здесь каждый житель указан в квадратных скобках, первое число — его достаток, второе число — его идентификационный номер. Таким образом, самый бедный житель имеет номер 3, самый богатый — номер 1, а средний — номер 4.

Исходный код к задаче 4

```
#include <iostream>
#include <vector>
#include <map>
#include <algorithm>
#include <fstream>
#include <inttypes.h>
using namespace std;
int main () {
    ifstream fi;
    ofstream fo;
    size_t n;
    double tmp;
    map <double, int> m;
    fi.open ("input.txt");
    fo.open ("output.txt");
    fi >> n;
    for (size_t i = 0; i < n; ++i)
    {
        fi >> tmp;
        m.emplace(tmp, i); // abuse map for sort
    }
    auto mid = m.begin();
    auto end = m.begin();
    advance(mid, n / 2);
    advance(end, n-1);
    fo << m.begin()->second+1 << " " << mid->second+1 << " " << end->second+1 << "\n";
    fi.close();
    fo.close();
    return 0;
}
```

Бенчмарк к задаче 4

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.093	4362240	98892	16
1	OK	0.015	3665920	30	7
2	OK	0.000	3682304	33	7
3	OK	0.000	3665920	1065	10
4	OK	0.000	3706880	3732	12
5	OK	0.015	3743744	14975	15
6	OK	0.015	3727360	14998	13
7	OK	0.062	3837952	28749	16

8	OK	0.031	3883008	34791	14
9	OK	0.046	3891200	38037	15
10	OK	0.046	3903488	38074	16
11	OK	0.031	3919872	39288	15
12	OK	0.046	3969024	48638	15
13	OK	0.046	4063232	50722	14
14	OK	0.046	4018176	52757	16
15	OK	0.046	4059136	58008	15
16	OK	0.046	4116480	66504	16
17	OK	0.046	4157440	71786	16
18	OK	0.062	4173824	72346	16
19	OK	0.046	4157440	73304	15
20	OK	0.046	4161536	76139	16
21	OK	0.093	4227072	83944	16
22	OK	0.062	4247552	85179	15
23	OK	0.078	4251648	86522	14
24	OK	0.078	4231168	89202	15
25	OK	0.062	4362240	98892	16

Задача 5 Секретарь Своп

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

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 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

Послесловие и предостережение

Семья секретаря Свопа занималась сортировками массивов, и именно с помощью перестановок пар элементов, как минимум с XII века, поэтому все Свопы владеют этим искусством в совершенстве. Мы не просим Вас произвести минимальную последовательность перестановок, приводящую к правильному ответу. Однако учтите, что для вывода слишком длинной последовательности у Вашего алгоритма может не хватить времени (или памяти — если выводимые строки хранятся в памяти перед выводом). Подумайте, что с этим можно сделать. Решение существует!

Исходный код к задаче 5

```
#include <iostream>
#include <vector>
#include <map>
#include <algorithm>
#include <fstream>
#include <inttypes.h>
using namespace std;
ifstream fi;
ofstream fo;
void quickSortR(long *a, long l, long r) {
    long s = l;
    long n = r;
    if (r-l < 1 )
        return;
    long temp, p;
    p = a[ (r+l)/2 ];
    do {
        while ( a[l] < p ) l++;
        while ( a[r] > p ) r--;
        if ( l < r ) {
            temp = a[l]; a[l] = a[r]; a[r] = temp;
            fo << "Swap elements at indices " << l+1 << " and " << r+1 << ".\n";
        }
        if ( l <= r ) {
            l++; r--;
        }
    } while ( l<=r );
    quickSortR(a, s, r);
    quickSortR(a, l, n);
}
int main () {
    size_t n;
    long a[5000];
    fi.open ("input.txt");
    fo.open ("output.txt");
    fi >> n;
    for (size_t i = 0; i < n; ++i)
    {
        fi >> a[i];
    }
    quickSortR(a, 0, n-1);
    fo << "No more swaps needed." << "\n";
    for (size_t i = 0; i < n; ++i)
    {
        fo << a[i] << " ";
    }
    fi.close();
    fo.close();
    return 0;
}
```

Бенчмарк к задаче 5

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.046	3706880	51993	635371
1	OK	0.015	3698688	14	138

2	OK	0.015	3670016	7	26
3	OK	0.015	3682304	12	31
4	OK	0.046	3706880	8	62
5	OK	0.000	3670016	10	64
6	OK	0.000	3694592	10	29
7	OK	0.000	3678208	29	48
8	OK	0.015	3661824	10	64
9	OK	0.015	3678208	10	64
10	OK	0.015	3698688	10	99
11	OK	0.015	3702784	10	64
12	OK	0.015	3682304	10	99
13	OK	0.000	3674112	50	209
14	OK	0.015	3670016	56	180
15	OK	0.015	3686400	57	76
16	OK	0.015	3694592	55	144
17	OK	0.015	3665920	75	304
18	OK	0.000	3686400	76	95
19	OK	0.000	3682304	78	202
20	OK	0.000	3682304	108	267
21	OK	0.015	3670016	107	125
22	OK	0.000	3694592	108	302
23	OK	0.000	3686400	948	5931
24	OK	0.015	3686400	947	965
25	OK	0.000	3678208	948	2622
26	OK	0.000	3678208	3720	31951
27	OK	0.031	3674112	3735	3752
28	OK	0.015	3674112	3722	10612
29	OK	0.000	3702784	8463	78937
30	OK	0.015	3678208	8441	8458

31	OK	0.015	3694592	8434	24177
32	OK	0.031	3694592	22822	252664
33	OK	0.015	3665920	22825	22841
34	OK	0.000	3698688	22877	66845
35	OK	0.031	3686400	51987	635371
36	OK	0.000	3694592	51940	51956
37	OK	0.015	3678208	51993	153402