

Задача 1 (Массивы и Структуры)

Обращение последовательности

Выведите последовательность в обратном порядке.

Ввод

Целое число n ($1 \leq n \leq 1000$) и последовательность из n целых чисел.

Вывод

Последовательность в обратном порядке.

Примеры

Вход	Выход
3 -9999 26 3	3 26 -9999
1 5	5
5 1 2 3 4 5	5 4 3 2 1

Задача 2 (Массивы и Структуры)

Больше среднего

Выведите элементы последовательности, которые больше среднего арифметического ее элементов.

Ввод

Целое число n ($1 \leq n \leq 1000$) и последовательность из n целых чисел.

Вывод

Элементы последовательности, значение которых больше среднего арифметического значения для этой последовательности, в порядке их ввода.

Примеры

Вход	Выход
3 1 2 3	3
1 5	5
5 1 2 3 4 5	4 5

Задача 3 (Массивы и Структуры)

Транспонирование квадратной матрицы

Транспонируйте небольшую квадратную матрицу.

Ввод

Целое число n (размер матрицы, $1 \leq n \leq 10$) и последовательность из $n \cdot n$ целых чисел (матрица построчно).

Вывод

Транспонированная матрица.

Примеры

Вход	Выход
2 1 2 3 4	 1 3 2 4
5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5	 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5 5 5 5

Задача 4 (Массивы и Структуры)

Транспонирование матрицы

Транспонируйте небольшую матрицу.

Ввод

Целые числа n и m (размеры матрицы, $1 \leq n, m \leq 10$) и последовательность из $n*m$ целых чисел (матрица построчно).

Вывод

Транспонированная матрица.

Примеры

Вход	Выход
2 2 1 2 3 4	1 3 2 4
5 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5	1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5 5 5 5
1 5 1 2 3 4 5	1 2 3 4 5

Задача 5 (Массивы и Структуры)

Повернуть картину

Считайте с экрана "картинку", нарисованную символами. Поверните ее на $\pi/2$ (против часовой стрелки) и выведите на экран. Рекомендуем для считывания символов использовать следующую функцию:

```
char c;  
cin.get(c);
```

Обратите внимание, что она будет считывать в том числе пробелы и окончания строк (`'\n'`).

Ввод

Два целых положительных числа n и m ($3 \leq n, m \leq 10$) - размеры "картинки". Набор из $n \cdot m$ символов, разбитых на строки по m символов.

Вывод

"Картинка", повернутая на $\pi/2$ (против часовой стрелки).

Примеры

Вход	Выход
<pre>3 3 *__ *__ *__</pre>	<pre>--- --- ***</pre>
<pre>3 10 ***** * * *****</pre>	<pre>*** * * * * * * * * * * * * * * * * ***</pre>
<pre>5 5 ***** * _ * * * * _ * *****</pre>	<pre>***** * _ * * * * _ * *****</pre>

Задача 6 (Массивы и Структуры)

Поиск максимальных чисел

Найдите определенное количество наибольших элементов массива.

Ввод

Целое положительное число n ($1 \leq n \leq 1000$) - размер массива. Последовательность из n целых чисел. Целое положительное число m ($1 \leq m \leq n$) - количество наибольших элементов.

Вывод

m наибольших элементов массива из n целых чисел, которые нужно вывести на экран в порядке возрастания.

Примеры

Вход	Выход
10 2 4 6 5 10 6 2 3 4 8 2	8 10
10 2 4 6 5 10 6 2 3 4 8 10	2 2 3 4 4 5 6 6 8 10
10 96 7 3 1 8 64 34 568 4 1 5	8 34 64 96 568

Задача 7 (Массивы и Структуры)

Поиск максимальных чисел

Найдите определенное количество наибольших элементов массива, сохранив порядок элементов.

Ввод

Целое положительное число n ($1 \leq n \leq 1000$) - размер массива. Последовательность из n целых чисел. Целое положительное число m ($1 \leq m \leq n$) - количество наибольших элементов.

Вывод

m наибольших элементов массива из n целых чисел, которые нужно вывести на экран в том порядке, в каком они шли в последовательности. При дублировании наименьшего из наибольших элементов включать в выводимую последовательность те, что идут раньше.

Примеры

Вход	Выход
10 2 4 6 5 10 6 2 3 4 8 6	4 6 5 10 6 8
10 2 4 6 5 10 6 2 3 4 9 2	10 9

Задача 8 (Массивы и Структуры)

Наибольший столбец

Найдите столбец прямоугольной матрицы, сумма элементов которого является наибольшей.

Ввод

Целые числа n и m (размеры матрицы, $1 \leq n, m \leq 100$) и последовательность из $n \cdot m$ целых чисел (матрица построчно).

Вывод

Номер столбца с наибольшей суммой элементов.

Примеры

Вход	Выход
2 2 1 2 3 4	1
5 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5	4
3 6 6 4 5 9 7 0 1 2 3 4 10 20 30 40 50 50 60 30	4

Задача 9 (Массивы и Структуры)

Сумма векторов

Напишите функцию, которая складывает два вектора.

```
struct Vector {  
    int x, y, z;  
};  
Vector sum(Vector a, Vector b);
```

Аргументы

Две структуры типа Vector.

Возвращаемое значение

Одна структура типа Vector, покомпонентная сумма аргументов.

Инфраструктура (код, который можно использовать при отладке)

Программа считывает с экрана данные для нескольких структур типа Vector и складывает их.

```
#include <iostream>  
using namespace std;  
struct Vector {  
    int x, y, z;  
};  
Vector sum(Vector a, Vector b);  
const int N = 10;  
int main()  
{  
    Vector res = {0, 0, 0};  
    for (int i=0; i<N; i++)  
    {  
        Vector tmp;  
        cin >>tmp.x >>tmp.y >>tmp.z;  
        res = sum(res, tmp);  
    }  
    cout <<res.x <<" " <<res.y <<" " <<res.z <<endl;  
    return 0;  
}
```

Примеры

Вход	Выход
0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0	4 5 4

Задача 10 (Массивы и Структуры)

Обработка протоссов

Напишите функцию, которая анализирует массив структур, содержащий данные о персонажах компьютерной игры, и находит тех, здоровье которых строго ниже определенного предела.

```
struct Protoss {
    unsigned long long int id;
    unsigned int health, shield, position[2];
    char name[100];
};

unsigned int count_wounded(Protoss* army, unsigned int n, unsigned int threshold);
```

Аргументы

Три параметра: массив протоссов, размер данного массива, порог допустимого здоровья.

Возвращаемое значение

Одно число - количество раненых протоссов.

Инфраструктура (код, который можно использовать при отладке)

Программа считывает с экрана данные для нескольких структур типа Protoss и находит количество раненых.

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

struct Protoss {
    unsigned long long int id;
    unsigned int health, shield, position[2];
    char name[100];
};

unsigned int count_wounded(Protoss* army, unsigned int n, unsigned int threshold);

int main()
{
    Protoss army[10];
    for (int i=0; i<10; i++)
    {
        army[i].id = i;
        cin >>army[i].health >>army[i].shield >>army[i].position[0] >army[i].position[1] >army[i].name;
    }

    cout <<count_wounded(army, 10, 100);
    cout <<endl;
    return 0;
}
```

Примеры

Вход	Выход
100 64 0 0 Fenix 65 366 100 100 Artanis 32 654 65 354 Dragoon_000 21 368 69 3 Dragoon_001 100 65 154 16 Tassadar 100 3 368 3697 Zeratul 68 153 354 386 Aldaris 100 368 758 6876 Alarac 100 6 57 9878 Rohana 100 687 6987 6987 Selendis	4

Задача sort

Вводится N целых чисел ($1 \leq N \leq 1000$). Выведите вначале отсортированные по возрастанию неотрицательные, потом отсортированные по возрастанию модуля отрицательные.

Формат входных данных

Целое число N - количество чисел. Далее N целых чисел, каждое с новой строки.

Формат выходных данных

Отсортированные числа, в одну строку, через пробел.

Примеры

Ввод	Вывод
5 2 -4 5 1 -2	1 2 5 -2 -4

Задача shop

В магазине продаются разные коробки конфет, всего их N ($1 \leq N \leq 1000$). Для каждой известны P (цена) и K (сколько конфет внутри). У вас есть S денег. Нужно купить максимальное количество **коробок**. Вывести, сколько всего получится коробок, и сколько внутри окажется конфет.

Формат входных данных

Целое число N - количество коробок. Далее N строк, в каждой целое число P (цена) и целое число K (сколько конфет внутри). Далее на новой строке целое число S .

Формат выходных данных

Два целых числа через пробел - сколько купили коробок и сколько внутри в сумме конфет.

Примеры

Ввод	Вывод
3 1 10 2 20 3 100 3	 2 30