

## Содержание

<b>Must have</b>	<b>2</b>
Задача 6А. Одномерный почтальон [0.8 sec, 256 mb]	2
Задача 6В. Одномерный финансист [0.2 sec, 256 mb]	3
Задача 6С. Точки и отрезки [0.2 sec, 256 mb]	4
<b>Обязательные задачи</b>	<b>5</b>
Задача 6D. Объединение прямоугольников [0.3 sec, 256 mb]	5
Задача 6Е. La cucaracha [0.2 sec, 256 mb]	6
Задача 6F. Добро пожаловать на сборы [2.0 sec, 256 mb]	7
Задача 6G. Q-я порядковая статистика [0.8 sec, 256 mb]	8
<b>Дополнительные задачи</b>	<b>9</b>
Задача 6H. Возня на круге [3.0 sec, 256 mb]	9
Задача 6I. Сортировка за линейное время [0.4 sec, 256 mb]	10
Задача 6J. Котята за круглым столом [0.8 sec, 256 mb]	11
Задача 6K. Ворчливые коровы [0.2 sec, 256 mb]	12

---

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

Обратите внимание на GNU C++ компиляторы с суффиксом `inc`, они позволяют пользоваться **дополнительной библиотекой**. Под ними можно сдать **вот это**.

## Must have

### Задача 6А. Одномерный почтальон [0.8 sec, 256 mb]

В деревне Печалька живут  $n$  человек, их домики расположены ровно на оси абсцисс. Домик  $i$ -го человека находится в точке  $x_i$ . В деревню приехал и хочет там поселиться почтальон. Координату своего домика  $y$  он хочет выбрать так, чтобы суммарное расстояние от него до всех жителей деревни было минимально возможным. То есть

$$\sum_{i=1}^n |y - x_i| \rightarrow \min$$

Вам дан массив  $x$  из  $n$  случайных целых чисел. Найдите точку  $y$ .

#### Формат входных данных

На первой строке число  $n$  ( $1 \leq n \leq 10^7$ ). На второй строке пара целых чисел  $a, b$  от 1 до  $10^9$ , используемая в генераторе случайных чисел.

```
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand24() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur >> 8; // число от 0 до 224 - 1.
5. }
6. unsigned int nextRand32() {
7.     unsigned int a = nextRand24(), b = nextRand24();
8.     return (a << 8) ^ b; // число от 0 до 232 - 1.
9. }
```

Элементы массива генерируются последовательно.  $x_i = \text{nextRand32}()$ ;

#### Формат выходных данных

Выведите одно число — минимальное суммарное расстояние от точки  $y$  до всех домиков.

#### Примеры

stdin	stdout
6	8510257371
239 13	

#### Замечание

Сгенерированный массив: 12, 130926, 3941054950, 2013898548, 197852696, 2753287507.

### Задача 6В. Одномерный финансист [0.2 sec, 256 mb]

В деревне Печалька живут  $n$  человек, их домики расположены ровно на оси абсцисс. Домик  $i$ -го человека находится в точке  $x_i$ . В деревню недавно заселился почтальон. Почтальон построил себе домик в такой точке  $y$ , что суммарное расстояние от него до всех жителей деревни было минимально возможным. А теперь в деревню приехал финансовый аналитик, который привык не только оптимизировать результат, но и оценивать риски. Посмотрев на опыт почтальона, аналитик заметил, что несмотря на то, что сумма минимальна, есть домики очень далеко от дома почтальона. Финансист учел это и свой дом хочет построить в такой точке  $z$ , что

$$\sum_{i=1}^n (z - x_i)^2 \rightarrow \min$$

С почтальоном финансист не дружит, поэтому расстояние до  $y$  в сумме не учитывается. Вам дан массив  $x$  из  $n$  случайных целых чисел. Найдите точку  $z$ .

#### Формат входных данных

На первой строке число  $n$  ( $1 \leq n \leq 10^7$ ). На второй строке пара целых чисел  $a, b$  от 1 до  $10^9$ , используемая в генераторе случайных чисел.

```
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand24() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur >> 8; // число от 0 до 224 - 1.
5. }
6. unsigned int nextRand32() {
7.     unsigned int a = nextRand24(), b = nextRand24();
8.     return (a << 8) ^ b; // число от 0 до 232 - 1.
9. }
```

Элементы массива генерируются последовательно.  $x_i = \text{nextRand32}()$ ;

#### Формат выходных данных

Выведите координату домика финансиста в виде несократимой дроби с положительным знаменателем.

#### Примеры

stdin	stdout
6	3368129374/3
230 10	

#### Замечание

Сгенерированный массив: 9, 1004452, 2338007883, 149525792, 917993446, 3329727166.

### Задача 6С. Точки и отрезки [0.2 sec, 256 mb]

На прямой в точках  $a_1, a_2, \dots, a_n$  (возможно, совпадающих) сидят  $n$  котят. На той же прямой лежат  $m$  отрезков  $[l_1, r_1], [l_2, r_2], \dots, [l_m, r_m]$ . Нужно для каждого отрезка узнать его *наполненность котятами* — сколько котят сидит на отрезке.

#### Формат входных данных

На первой строке  $n$  и  $m$  ( $1 \leq n, m \leq 10^5$ ). На второй строке  $n$  целых чисел  $a_i$  ( $0 \leq a_i \leq 10^9$ ). Следующие  $m$  строк содержат пары целых чисел  $l_i, r_i$  ( $0 \leq l_i \leq r_i \leq 10^9$ ).

#### Формат выходных данных

Выведите  $m$  целых чисел.  $i$ -е число — *наполненность котятами*  $i$ -го отрезка.

#### Примеры

stdin	stdout
4 5	3 4 3 1 0
1 2 3 5	
1 4	
1 5	
2 5	
3 3	
0 0	

## Обязательные задачи

### Задача 6D. Объединение прямоугольников [0.3 сек, 256 mb]

На плоскости задано  $N$  прямоугольников с вершинами в точках с целыми координатами и сторонами, параллельными осям координат. Необходимо найти площадь их объединения.

#### Формат входных данных

В первой строке входного файла указано число  $N$  ( $0 \leq N \leq 1500$ ). В следующих  $N$  строках заданы по 4 целых числа  $x_1, y_1, x_2, y_2$  — сначала координаты левого нижнего угла прямоугольника, потом правого верхнего ( $0 \leq x_1 \leq x_2 \leq 10^9, 0 \leq y_1 \leq y_2 \leq 10^9$ ). Обратите внимание, что прямоугольники могут вырождаться в отрезки и даже в точки.

#### Формат выходных данных

В выходной файл выведите единственное число — ответ на задачу.

#### Пример

stdin	stdout
3 1 1 3 5 5 2 7 4 2 4 6 7	23

### Задача 6Е. La cucaracha [0.2 sec, 256 mb]

Каждую полночь в квартире ученого Васи начинается ужас. Сотни . . . , о нет! ТЫСЯЧИ тараканов вылазят из каждой дырки к его обеденному столу, уничтожая все крошки и объедки! Вася ненавидит тараканов. Он очень долго думал и сделал Супер-ловушку, которая привлекает всех тараканов в большой зоне после активации. Он планирует активировать ловушку сегодня ночью. Но есть проблема. Эта очень эффективная ловушка с её очень большой зоной работы поглощает огромное количество энергии. Так что, Вася планирует минимизировать время работы этой ловушки. Он собрал информацию о всех местах, в которых живут тараканы. Также он заметил, что все тараканы двигаются только по линиям его скатерти с постоянной скоростью (мы можем предположить, что эта скорость равна 1, так что таракан расположенный в одной из секций, может за 1 единицу времени переместиться на любую соседнюю секцию (по вертикали или горизонтали)). Вася решил активировать его ловушку в одной из секций. Когда ловушка активирована, все тараканы будут двигаться к секции, содержащей ловушку, так быстро, как только смогут. Поэтому в любой момент времени после активации тараканы двигаются к секции, в которой находится ловушка, максимально уменьшая расстояние до неё. Если есть два пути с одинаковым расстоянием, то таракан выберет любой. Напишите программу для Васи, которая выбирает секцию, минимизирующую время, необходимое для уничтожения всех тараканов. Конечно, ваша программа будет считать, что скатерть будет плоскостью с декартовой системой координат и секции — точки с целыми координатами.

#### Формат входных данных

В первой строке входного файла содержится число мест, в которых живут тараканы  $N$  ( $1 \leq N \leq 10000$ ). Следующие  $N$  строк содержат  $x$  и  $y$  — координаты мест, в которых живут тараканы (целые числа не больше  $10^9$  по абсолютному значению).

#### Формат выходных данных

Вам необходимо вывести только два целых числа  $x$  и  $y$ , не превосходящие по модулю  $10^9$ , — координаты секции, которая минимизирует время работы. Если есть более одно решение — выведите любое из них.

#### Пример

stdin	stdout
2	2 2
1 1	
3 3	

### Задача 6F. Добро пожаловать на сборы [2.0 сек, 256 mb]

Как известно, всенародные сборы по программированию и информированию собирают участников со всей страны Б.! Поэтому организаторы сборов решили в следующий раз выбрать место проведения таким образом, чтобы минимизировать суммарное время, которое потребуется всем участникам для того, чтобы добраться до выбранного места.

Для простоты будем считать, что страна Б. представляет собой плоскость со стандартной декартовой системой координат, участники изначально расположены в целочисленных точках этой плоскости, из точки  $(x, y)$  он может перейти в точку  $(x_1, y_1)$  за время  $\max(|x - x_1|, |y - y_1|)$ . Необходимо выбрать точку с рациональными координатами для места проведения сборов так, чтобы минимизировать **суммарное** время, потраченное участниками на дорогу до места сборов.

Вам даны  $n$  групп участников.  $i$ -я группа расположена в точке  $x_i, y_i$  и имеет численность  $w_i$ . Найдите минимальное суммарное время в виде несократимой дроби.

#### Формат входных данных

В первой строке записано единственное число  $t$  ( $1 \leq t \leq 10^5$ ) — количество тестовых примеров. Каждый тестовый пример описывается следующим образом:

В первой строке описания записано целое число  $n$  — количество участников сборов. В следующих  $n$  строках описываются места, в которых изначально расположены участники. Каждое описание состоит из трёх целых чисел  $x, y$  ( $0 \leq x, y \leq 10^9$ ) и  $w$  ( $1 \leq w \leq 10^9$ ) и говорит о том, что в точке с координатами  $(x, y)$  изначально расположены  $w$  участников. Гарантируется, что сумма всех  $w$  в каждом тестовом примере не превосходит  $10^9$ .

Сумма всех  $n$  во входных данных не превосходит 3 000 000.

#### Формат выходных данных

В первой строке выведите число  $t$  — количество тестовых примеров, в каждой из следующих  $t$  строк выведите ответ на соответствующий тест — числитель и знаменатель несократимой дроби.

#### Пример

stdin	stdout
2	2
4	66 1
3 1 10	4 1
7 0 9	
5 6 8	
4 3 1	
3	
1 1 1	
2 2 2	
3 3 3	

### Задача 6G. Q-я порядковая статистика [0.8 sec, 256 mb]

Вам дан массив из  $n$  случайных целых чисел. Ваша задача — отсортировать массив и вывести  $q$ -е число в получившемся порядке.

#### Формат входных данных

На первой строке числа  $n, q$ . ( $1 \leq q \leq n \leq 10^7$ ). На второй строке пара целых чисел  $a, b$  от 1 до  $10^9$ , используемая в генераторе случайных чисел.

```
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand24() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur >> 8; // число от 0 до  $2^{24} - 1$ .
5. }
6. unsigned int nextRand32() {
7.     unsigned int a = nextRand24(), b = nextRand24();
8.     return (a << 8) ^ b; // число от 0 до  $2^{32} - 1$ .
9. }
```

Элементы массива генерируются последовательно.  $x_i = \text{nextRand32}()$ ;

#### Формат выходных данных

Выведите ответ на запрос.

#### Примеры

stdin	stdout
6 3	197852696
239 13	

#### Замечание

Сгенерированный массив: 12, 130926, 3941054950, 2013898548, 197852696, 2753287507.  
Внимание: нельзя пользоваться стандартными функциями, например, `nth_element`.



## Дополнительные задачи

### Задача 6Н. Возня на круге [3.0 сек, 256 mb]

Представьте себе клетчатую окружность, состоящую из  $L$  клеток. Клетки нумеруются целыми числами от 1 до  $L$ . Некоторые  $N$  клеток закрашены. Окружность можно разрезать между любыми двумя клетками. Всего существует  $L$  различных разрезов. Получившаяся полоска делится на  $K$  равных частей ( $L$  кратно  $K$ ).  $\forall i$  определим  $f_i$ , как количество закрашенных клеток в  $i$ -й части. Вам нужно найти такой разрез, что

$$F = \max_{i=1..K} f_i - \min_{i=1..K} f_i$$

минимально возможно. Кроме того, вам нужно найти количество разрезов, на которых достигается максимум.

#### Формат входных данных

На первой строке целые числа  $L, N, K$ , ( $K \geq 2$ ). Гарантируется, что  $L$  делится на  $K$ . На второй строке  $N$  различных целых чисел от 1 до  $L$  — номера закрашенных клеток.

#### Формат выходных данных

На первой строке выведите минимальное значение  $F$  и количество разрезов с таким  $F$ . На второй строке выведите любое  $x$  от 1 до  $L$ , что, если разрезать окружность между клетками  $x$  и  $x + 1$ , а после чего посчитать  $F$ , то получится максимальное значение.

**Ограничения:**  $N \leq 500\,000, L \leq 10^{18}$ .

#### Примеры

stdin	stdout
10 1 5 7	1 10 3
10 4 2 3 7 5 2	0 4 3

#### Пояснение к примеру

В первом примере подойдет любой разрез.

Во втором примере разрезать можно после 3, 4, 8, 9 клеток.

### Задача 61. Сортировка за линейное время [0.4 сек, 256 mb]

Дан массив случайных целых чисел, нужно отсортировать его.

#### Формат входных данных

На первой строке количество тестов  $t$  ( $1 \leq t \leq 200$ ) и число  $n$  ( $1 \leq n \leq 50\,000$ ) — размер массива в каждом из тестов. На второй строке пара целых чисел  $a, b$  от 1 до  $10^9$ , используемая в генераторе случайных чисел.

```
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand24() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur >> 8; // число от 0 до 224 - 1.
5. }
6. unsigned int nextRand32() {
7.     unsigned int a = nextRand24(), b = nextRand24();
8.     return (a << 8) ^ b; // число от 0 до 232 - 1.
9. }
```

Тесты генерируются последовательно.

Элементы массива генерируются последовательно.  $x_i = \text{nextRand32}()$ ;

#### Формат выходных данных

Для каждого теста выведите на отдельной строке  $\sum_{i=1}^n x_i \cdot i$ .

#### Примеры

stdin	stdout
1 6	46062181379
239 13	

#### Замечание

Сгенерированный массив: 12, 130926, 3941054950, 2013898548, 197852696, 2753287507.

В этой задаче очень небольшой запас по времени. Если она не сдается, это нормально.

Предполагается решение через или BucketSort, RadixSort.

### Задача 6J. Котятa за круглым столом [0.8 sec, 256 mb]

Котятa собрались за круглым столом, чтобы отпраздновать успешное замяукивание. У них есть блюдечки с молоком, которые нужно расставить вдоль границы стола. Каждый котёнок будет лакать молоко из блюдца, которое ближе всего к нему (расстояние измеряется как длина дуги). Требуется выбрать такие целочисленные позиции для блюдец (длина окружности — целое число, на окружности есть точки с координатами 0, 1, 2 и т.д.), что максимальное расстояние среди всех котят до нужного блюдца будет минимальным.

#### Формат входных данных

В первой строке ввода записаны целые числа  $N$ ,  $K$  и  $L$  — количество котят и блюдец, а также длина окружности стола. В следующей строке записано  $N$  целых чисел от 0 до  $(L - 1)$  в возрастающем порядке — позиции котят.  $1 \leq N, K \leq 500\,000$ ,  $1 \leq L \leq 1\,000\,000\,000$ .

#### Формат выходных данных

В первой строке выведите минимально возможное значение максимального расстояния. Во второй выведите  $K$  целых чисел от 0 до  $(L - 1)$  — позиции блюдец в любом из оптимальных решений.

#### Примеры

stdin	stdout
4 2 1000000	5000
0 67000 68000 77000	0 72000

### Задача 6К. Ворчливые коровы [0.2 сек, 256 mb]

У фермера Джона  $N$  коров. Каждый вечеро они выстраиваются в очередь, чтобы их подоили. У каждой коровы есть показатель ворчливости — целое число. Показатели ворчливости всех коров попарно различны.

Ворчливые коровы могут повредить оборудование фермы, поэтому фермер Джон хочет переупорядочить их так, чтобы доить коров в порядке возрастания их ворчливости. Фермер может выбрать двух любых коров (не обязательно стоящих рядом) с ворчливостями  $X$  и  $Y$  и поменять их местами за время  $X + Y$ . Эту операцию можно повторять последовательно сколько угодно раз.

Помогите фермеру Джону вычислить минимальное время, которое потребуется для того, чтобы переставить коров в очереди таким образом, чтобы доить их в порядке возрастания ворчливости.

#### Формат входных данных

В первой строке ввода задано целое число  $N$  — количество коров ( $1 \leq N \leq 10\,000$ ). Следующие  $N$  строк содержат по одному числу каждая. Это ворчливости коров в порядке очереди на дойку. Гарантируется, что ворчливости — попарно различные целые числа от 1 до 100 000.

#### Формат выходных данных

Выведите одно число — минимальное время, за которое можно переупорядочить коров в порядке возрастания ворчливости.

#### Пример

stdin	stdout
3	7
2	
3	
1	

#### Пояснение к примеру

В примере три коровы. Изначально первой стоит корова с ворчливостью 2, второй — с ворчливостью 3, а третьей — с ворчливостью 1.

Фермер Джон может сначала поменять коров с ворчливостями 1 и 3 за время  $1+3 = 4$  и из порядка 2 3 1 получить порядок 2 1 3. Затем фермер может поменять коров с ворчливостями 1 и 2 за время  $1 + 2 = 3$  и получить требуемый порядок 1 2 3.