

## Содержание

<b>Обязательные задачи</b>	<b>3</b>
Задача 04А. Быстрый поиск в массиве [0.2 sec, 256 mb]	3
Задача 04В. Для любителей статистики [0.2 sec, 256 mb]	4
Задача 04С. Веребочки [0.2 sec, 256 mb]	5
Задача 04D. Q-я порядковая статистика [0.6 sec, 256 mb]	6
Задача 04Е. Минимальное покрытие [0.2 sec, 256 mb]	7
Задача 04F. Точки и отрезки [0.2 sec, 256 mb]	8
Задача 04G. Том Сойер и его друзья [0.2 sec, 256 mb]	9
<b>Дополнительные задачи</b>	<b>10</b>
Задача 04Н. Китайские часы [0.2 sec, 256 mb]	10
Задача 04I. K-Best [0.2 sec, 256 mb]	11
Задача 04J. Лифт [0.2 sec, 256 mb]	12
Задача 04К. Долгий путь [0.5 sec, 256 mb]	13

---

Во всех задачах можно использовать `std::sort`.

### **Пример работы с файлами.**

Если вы не умеете читать/выводить данные, или открывать файлы, воспользуйтесь примерами. <http://acm.math.spbu.ru/~sk1/algo/sum/>

### **Пример работы с файлами.**

В некоторых задачах большой ввод и вывод. Про ввод-вывод в C++:

[http://acm.math.spbu.ru/~sk1/algo/input-output/cpp\\_common.html](http://acm.math.spbu.ru/~sk1/algo/input-output/cpp_common.html)

Имеет смысл пользоваться супер быстрым вводом-выводом. Две версии:

[http://acm.math.spbu.ru/~sk1/algo/input-output/io\\_export.cpp.html](http://acm.math.spbu.ru/~sk1/algo/input-output/io_export.cpp.html)

[http://acm.math.spbu.ru/~sk1/algo/input-output/fread\\_write\\_export.cpp.html](http://acm.math.spbu.ru/~sk1/algo/input-output/fread_write_export.cpp.html)

### **Выделение памяти.**

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

<http://acm.math.spbu.ru/~sk1/algo/memory.cpp.html>

## Обязательные задачи

### Задача 04А. Быстрый поиск в массиве [0.2 сек, 256 mb]

Дан массив из  $N$  целых чисел. Все числа от  $-10^9$  до  $10^9$ .

Нужно уметь отвечать на запросы вида “Сколько чисел имеют значения от  $L$  до  $R$ ?”.

#### Формат входных данных

Число  $N$  ( $1 \leq N \leq 10^5$ ). Далее  $N$  целых чисел.

Затем число запросов  $K$  ( $1 \leq K \leq 10^5$ ).

Далее  $K$  пар чисел  $L, R$  ( $-10^9 \leq L \leq R \leq 10^9$ ) — собственно запросы.

#### Формат выходных данных

Выведите  $K$  чисел — ответы на запросы.

#### Пример

find3.in	find3.out
5	5 2 2 0
10 1 10 3 4	
4	
1 10	
2 9	
3 4	
2 2	

**Задача 04В. Для любителей статистики [0.2 sec, 256 mb]**

Вы никогда не задумывались над тем, сколько человек за год перевозят трамваи города с десятиллионным населением, в котором каждый третий житель пользуется трамваем по два раза в день?

Предположим, что на планете Земля  $n$  городов, в которых есть трамваи. Любители статистики подсчитали для каждого из этих городов, сколько человек перевезено трамваями этого города за последний год. Из этих данных была составлена таблица, в которой города были отсортированы по алфавиту. Позже выяснилось, что для статистики названия городов несущественны, и тогда их просто заменили числами от 1 до  $n$ . Поисковая система, работающая с этими данными, должна уметь быстро отвечать на вопрос, есть ли среди городов с номерами от  $l$  до  $r$  такой, что за год трамваи этого города перевезли ровно  $x$  человек. Вам предстоит реализовать этот модуль системы.

**Формат входных данных**

В первой строке дано целое число  $n$ ,  $0 < n < 70\,000$ . В следующей строке приведены статистические данные в виде списка целых чисел через пробел,  $i$ -е число в этом списке — количество человек, перевезенных за год трамваями  $i$ -го города. Все числа в списке положительны и не превосходят  $10^9 - 1$ . В третьей строке дано количество запросов  $q$ ,  $0 < q < 70\,000$ . В следующих  $q$  строках перечислены запросы. Каждый запрос — это тройка целых чисел  $l$ ,  $r$  и  $x$ , записанных через пробел ( $1 \leq l \leq r \leq n$ ,  $0 < x < 10^9$ ).

**Формат выходных данных**

Выведите строку длины  $q$ , в которой  $i$ -й символ равен 1, если ответ на  $i$ -й запрос утвердителен, и 0 в противном случае.

**Пример**

queries.in	queries.out
5	10101
123 666 314 666 434	
5	
1 5 314	
1 5 578	
2 4 666	
4 4 713	
1 1 123	

### Задача 04С. Веревочки [0.2 sec, 256 mb]

С утра шел дождь, и ничего не предвещало беды. Но к обеду выглянуло солнце, и в лагерь заглянула СЭС. Пройдя по всем домикам и корпусам, СЭС вынесла следующий вердикт: бельевые веревки в жилых домиках не удовлетворяют нормам СЭС. Как выяснилось, в каждом домике должно быть ровно по одной бельевой веревке, и все веревки должны иметь одинаковую длину. В лагере имеется  $N$  бельевых веревок и  $K$  домиков. Чтобы лагерь не закрыли, требуется так нарезать данные веревки, чтобы среди получившихся веревочек было  $K$  одинаковой длины. Размер штрафа обратно пропорционален длине бельевых веревок, которые будут развешены в домиках. Поэтому начальство лагеря стремится максимизировать длину этих веревочек.

#### Формат входных данных

В первой строке заданы два числа —  $N$  ( $1 \leq N \leq 10\,001$ ) и  $K$  ( $1 \leq K \leq 10\,001$ ). Далее в каждой из последующих  $N$  строк записано по одному числу — длине очередной бельевой веревки. Длина веревки задана в сантиметрах. Все длины лежат в интервале от 1 сантиметра до 100 километров включительно.

#### Формат выходных данных

В выходной файл следует вывести одно целое число — максимальную длину веревочек, удовлетворяющую условию, в сантиметрах. В случае, если лагерь закроют, выведите 0.

#### Пример

ropes.in	ropes.out
4 11	200
802	
743	
457	
539	

### Задача 04D. Q-я порядковая статистика [0.6 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

qstat.in	qstat.out
6 3	197852696
239 13	

#### Замечание

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

**Внимание:** нельзя пользоваться стандартными функциями, например, `nth_element`.

### Задача 04Е. Минимальное покрытие [0.2 сек, 256 mb]

На прямой задано некоторое множество отрезков с целочисленными координатами концов  $[L_i, R_i]$ . Выберите среди данного множества подмножество отрезков, целиком покрывающее отрезок  $[0, M]$ , ( $M$  — натуральное число), содержащее наименьшее число отрезков.

#### Формат входных данных

В первой строке указана константа  $M$  ( $1 \leq M \leq 5\,000$ ). В каждой последующей строке записана пара чисел  $L_i$  и  $R_i$  ( $|L_i|, |R_i| \leq 50\,000$ ), задающая координаты левого и правого концов отрезков. Список завершается парой нулей. Общее число отрезков не превышает 100 000.

#### Формат выходных данных

В первой строке выходного файла выведите минимальное число отрезков, необходимое для покрытия отрезка  $[0, M]$ . Далее выведите список покрывающего подмножества, упорядоченный по возрастанию координат левых концов отрезков. Список отрезков выводится в том же формате, что и во входе. Завершающие два нуля выводить не нужно.

Если покрытие отрезка  $[0, M]$  исходным множеством отрезков  $[L_i, R_i]$  невозможно, то следует вывести единственную фразу "No solution".

#### Примеры

cover.in	cover.out
1 -1 0 -5 -3 2 5 0 0	No solution
1 -1 0 0 1 0 0	1 0 1

**Задача 04F. Точки и отрезки [0.2 sec, 256 mb]**

Дано  $n$  отрезков на числовой прямой и  $m$  точек на этой же прямой. Для каждой из данных точек определите, скольким отрезкам она принадлежит. Точка  $x$  считается принадлежащей отрезку с концами  $a$  и  $b$ , если выполняется двойное неравенство  $\min(a, b) \leq x \leq \max(a, b)$ .

**Формат входных данных**

Первая строка содержит два целых числа  $n$  ( $1 \leq n \leq 10^5$ ) — число отрезков и  $m$  ( $1 \leq m \leq 10^5$ ) — число точек. В следующих  $n$  строках записаны по два целых числа  $a_i$  и  $b_i$  — координаты концов соответствующего отрезка. В последней строке записаны  $m$  целых чисел — координаты точек. Все числа во входном файле не превосходят по модулю  $10^9$ .

**Формат выходных данных**

В выходной файл выведите  $m$  чисел — для каждой точки выведите количество отрезков, в которых она содержится.

**Примеры**

segments.in	segments.out
2 2 0 5 7 10 1 6	1 0
1 3 -10 10 -100 100 0	0 0 1

### Задача 04G. Том Сойер и его друзья [0.2 sec, 256 mb]

Друзья Тома Сойера по очереди красят забор разными красками. Каждый из них красит несколько идущих подряд секций забора в определенный цвет, при этом используемые цвета могут повторяться. Новая краска ложится поверх старой. Для каждой краски вычислите количество секций, которые будут покрашены этой краской после того, как все друзья закончат работу.

#### Формат входных данных

В первой строке входного файла содержатся два целых числа:  $N$  ( $1 \leq N \leq 10^9$ ) и  $K$  ( $1 \leq K \leq 50\,000$ ) — количество секций в заборе и количество различных красок соответственно.

Во второй строке содержится единственное число  $M$  ( $0 \leq M \leq 50\,000$ ) — количество друзей Тома Сойера.

Далее следуют  $M$  строк: в  $i$ -ой строке содержится информация о работе друга, который красил забор  $i$ -ым по счету, а именно 3 целых числа  $c_i, l_i, r_i$  ( $1 \leq c_i \leq K, 1 \leq l_i \leq r_i \leq N$ ) — номер краски, которую использовал  $i$ -й друг, номер первой и номер последней покрашенной секции соответственно.

#### Формат выходных данных

Выведите в единственную строку выходного файла  $K$  целых чисел:  $i$ -ое число должно быть равно количеству секций, покрашенных  $i$ -й краской.

#### Пример

paint.in	paint.out
5 3 4 1 3 4 2 4 5 3 2 3 1 5 5	1 1 2
5 3 3 1 1 5 2 2 4 1 3 3	3 2 0

#### Замечание

**Внимание:** в этой задаче нельзя пользоваться деревом отрезков. Зато можно использовать STL.

## Дополнительные задачи

### Задача 04Н. Китайские часы [0.2 sec, 256 mb]

Когда программист Вова был в Китае, он обнаружил, что российские часы “Заря” продаются там в десять раз дешевле, чем в России. Вова решил провернуть выгодное дело и закупил огромную партию с тем, чтобы привезти часы на родину и продать их за полцены (то есть впятеро дороже, чем купил). Но дома обнаружилось, что многие часы идут вразнобой, к тому же, от малейшего толчка часы останавливаются (либо наоборот, начинают идти). Очевидно, это были не настоящие часы фирмы “Заря”, а их точные копии. Чтобы быстро продать всю партию часов, Вова хочет поставить их все на одинаковое время (тогда будет неважно, правильное это время или нет — можно будет сказать, что это местное время завода-изготовителя) и перед открыванием крышки просто встряхнуть чемодан, чтобы часы одновременно пошли.

Для установки времени на часах есть заводная головка, вращение которой позволяет быстро поворачивать стрелки, при этом часовая стрелка крутится в 60 раз медленнее минутной, а минутная — в 60 раз медленнее секундной. Один оборот заводной головки поворачивает секундную стрелку на полный оборот, и хотя на такой поворот уходит всего одна секунда, чтобы изменить время на 6 часов, потребуется 6 минут. Вращать заводную головку и стрелки можно только по часовой стрелке, чтобы не повредить хрупкий механизм часов. Помогите Вове минимизировать усилия по предпродажной подготовке часов, выбрав время, которое будет установлено на всех часах.

#### Формат входных данных

В первой строке содержится одно целое число  $n$  ( $1 \leq n \leq 50\,000$ ) — количество часов в партии. Далее следует  $n$  строк,  $(i + 1)$ -я строка ввода содержит время на  $i$ -х часах в формате  $h : mm : ss$ , где целое число  $h$  ( $1 \leq h \leq 12$ ) означает час, а двухразрядные целые числа  $mm$  и  $ss$  ( $00 \leq mm, ss \leq 59$ ) — минуты и секунды соответственно.

#### Формат выходных данных

Выведите время, которое нужно установить на всех часах, в формате, указанном выше.

#### Пример

watches.in	watches.out
3	12:10:01
11:30:00	
12:10:01	
6:10:18	

**Задача 041. K-Best [0.2 sec, 256 mb]**

У Демьяны есть  $n$  драгоценностей. Каждая из драгоценностей имеет ценность  $v_i$  и вес  $w_i$ . С тех пор, как её мужа Джонни уволили в связи с последним финансовым кризисом, Демьяна решила продать несколько драгоценностей. Для себя она решила оставить лишь  $k$  лучших. Лучших в смысле максимизации достаточно специфического выражения: пусть она оставила для себя драгоценности номер  $i_1, i_2, \dots, i_k$ , тогда максимальной должна быть величина

$$\frac{\sum_{j=1}^k v_{i_j}}{\sum_{j=1}^k w_{i_j}}$$

Помогите Демьяне выбрать  $k$  драгоценностей требуемым образом.

**Формат входных данных**

На первой строке  $n$  и  $k$  ( $1 \leq k \leq n \leq 100\,000$ ).

Следующие  $n$  строк содержат пары целых чисел  $v_i, w_i$  ( $0 \leq v_i \leq 10^6, 1 \leq w_i \leq 10^6$ , сумма всех  $v_i$  не превосходит  $10^7$ , сумма всех  $w_i$  также не превосходит  $10^7$ ).

**Формат выходных данных**

Выведите  $k$  различных чисел от 1 до  $n$  — номера драгоценностей. Драгоценности нумеруются в том порядке, в котором перечислены во входных данных. Если есть несколько оптимальных ответов, выведите любой.

**Пример**

kbest.in	kbest.out
3 2	1 2
1 1	
1 2	
1 3	

### Задача 04J. Лифт [0.2 sec, 256 mb]

Высокое здание, состоящее из  $N$  этажей, оснащено только одним лифтом. Парковка находится ниже фундамента здания, что соответствует одному этажу ниже первого. Этажи пронумерованы от 1 до  $N$  снизу вверх. Про каждый этаж известно количество человек, желающих спуститься на лифте на парковку. Пусть для  $i$ -го этажа эта величина равна  $A_i$ . Известно, что лифт не может перевозить более  $C$  человек одновременно, а также то, что на преодоление расстояния в один этаж (не важно вверх или вниз) ему требуется  $P$  секунд. Какое наибольшее количество человек лифт может перевезти на парковку за  $T$  секунд, если изначально он находится на уровне парковки?

#### Формат входных данных

В первой строке входного файла содержатся целые числа  $N, C, P, T$  ( $1 \leq N \leq 100$ ,  $1 \leq C \leq 10^9$ ,  $1 \leq P \leq 10^9$ ,  $1 \leq T \leq 10^9$ ). Вторая строка содержит последовательность  $N$  целых чисел  $A_1, A_2, \dots, A_N$  ( $0 \leq A_i \leq 10^9$ ). Сумма всех значений последовательности не превосходит  $10^9$ .

#### Формат выходных данных

Выведите наибольшее количество человек, которое лифт успеет перевезти на парковку.

#### Пример

lift.in	lift.out
4 5 2 15 0 1 2 3	3
4 5 2 18 0 1 2 3	5
3 2 1 9 1 1 1	3

### Задача 04К. Долгий путь [0.5 sec, 256 mb]

Участники китайских сборов по спортивной ходьбе идут по дороге от железнодорожной станции к месту проведения сборов. Упрощенно, дорога представляет собой прямую, причем бесконечную. В некоторый момент времени выяснилось, что все участники разбились на  $n$  групп по  $k_i$  людей в каждой. Каждая из групп в этот момент времени находится в точке с координатой  $x_i$ . Известно, что если в группе  $D$  людей, то группа движется со скоростью  $100/D$  в единицу времени (группа движется вправо — в направлении возрастания координаты). Если какая-то группа догоняет другую, то они сливаются, и соответствующим образом меняется их скорость (может слиться более двух групп одновременно). Так как дорога бесконечна, то в какой-то момент времени наступит такая ситуация, что уже ни одного слияния не произойдет. Вам требуется определить окончательное количество групп и количества людей в них.

#### Формат входных данных

В первой строке входного файла находится число  $n$  ( $1 \leq n \leq 10^5$ ). Далее следует  $n$  строк, в каждой из которых написано количество людей в соответствующей группе  $k_i$  и ее координата  $x_i$  ( $x_i$  — вещественные числа с не более чем тремя знаками после запятой и не превосходят  $10^4$  по модулю,  $1 \leq k_i \leq 100$ , все координаты различны).

#### Формат выходных данных

В первой строке выведите количество групп  $m$ , которые остались. Во второй строке выведите  $m$  чисел — количества людей в этих группах, в произвольном порядке.

#### Пример

longpath.in	longpath.out
4	2
1 0	5 5
2 9000	
4 1	
3 10000	