

## Задача А. Двоичный поиск

Имя входного файла: bins.in  
Имя выходного файла: bins.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

### Формат входных данных

В первой строке входного файла содержатся числа  $N$  и  $K$  ( $0 \leq N, K \leq 10^5$ ). Во второй строке находится  $N$  чисел первого массива, а в третьей строке  $K$  чисел второго массива, каждое из которых по модулю не превосходящие  $2 \cdot 10^9$ . Числа первого массива отсортированы по неубыванию.

### Формат выходных данных

В выходной файл для каждого из  $K$  чисел вывести в отдельную строку «YES», если это число встречается в первом массиве и «NO» в противном случае.

### Пример

bins.in	bins.out
5 4	YES
1 4 5 8 9	NO
5 6 1 9	YES
	YES

## Задача В. Минимальное покрытие

Имя входного файла: cover.in  
Имя выходного файла: cover.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

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

### Формат входных данных

В первой строке указана константа  $M$  ( $1 \leq M \leq 5000$ ). В каждой последующей строке записана пара чисел  $L_i$  и  $R_i$  ( $|L_i|, |R_i| \leq 50000$ ), 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 вывести 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 С. Корень кубического уравнения

Имя входного файла: cubroot.in  
Имя выходного файла: cubroot.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дано кубическое уравнение  $ax^3+bx^2+cx+d=0$  ( $a \neq 0$ ). Известно, что у этого уравнения есть ровно один корень. Требуется его найти.

### Формат входных данных

Во входном файле через пробел записаны четыре целых числа:  $-1000 \leq a, b, c, d \leq 1000$ .

### Формат выходных данных

Выведите единственный корень уравнения с точностью не менее 6 знаков после десятичной точки.

### Пример

cubroot.in	cubroot.out
1 -3 3 -1	1
-1 -6 -12 -7	-1.000000

## Задача D. Поляна Дров

Имя входного файла: forest.in  
Имя выходного файла: forest.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Маленький мальчик Ферма́ живет в деревне. Наступают холодные времена, поэтому бабушка попросила мальчика сходить в лес, чтобы собрать дров. В лесу около деревни, в которой живет Ферма, находится волшебная Поляна Дров, на которой всегда лежат дрова, и никогда не кончаются. Естественно, Ферма должен пойти именно туда.

Единственная проблема заключается в том, что идти до Поляны не очень близко, тем более что скорость передвижения по лесу намного меньше, чем скорость передвижения по полю, в котором находится деревня.

- Деревня находится в точке с координатами  $(0, 1)$ .
- Поляна находится в точке с координатами  $(1, 0)$ .
- Граница между лесом и полем — горизонтальная прямая  $y = a$ , где  $a$  — некоторое число ( $0 \leq a \leq 1$ ).
- Скорость передвижения по полю составляет  $V_p$ , скорость передвижения по лесу —  $V_f$ . Вдоль границы можно двигаться как по лесу, так и по полю.

Найдите точку, в которой мальчик Ферма должен войти в лес, чтобы дойти до Поляны Дров как можно быстрее.

### Формат входных данных

В первой строке входного файла содержатся два положительных целых числа —  $V_p$  и  $V_f$  ( $1 \leq V_p, V_f \leq 10^5$ ). Во второй строке содержится единственное вещественное число — координата по оси  $Oy$  границы между лесом и полем  $a$  ( $0 \leq a \leq 1$ ).

### Формат выходных данных

В единственной строке выходного файла выведите вещественное число с точностью не менее 8 знаков после запятой — координата по оси  $Ox$  точки, в которой мальчик Ферма должен войти в лес.

### Пример

forest.in	forest.out
5 3 0.4	0.783310604
5 5 0.5	0.500000000

Указание. Для обеспечения требуемой точности используйте тип данных `extended` в Паскале и `long double` в Си.

## Задача Е. Объединение отрезков

Имя входного файла: `merge.in`  
Имя выходного файла: `merge.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Решая задачу из контрольной по математике, Вася получил ответ в виде объединения  $N$  отрезков  $[L_i, R_i]$  на числовой прямой. Однако, некоторые из этих отрезков могут пересекаться друг с другом, что не слишком нравится Васе. Ваша задача — представить Васин ответ в виде объединения минимального количества отрезков.

### Формат входных данных

В первой строке указано число  $N$  ( $1 \leq N \leq 50000$ ). В следующих  $N$  строках перечислены пары чисел  $L_i$  и  $R_i$  ( $|L_i|, |R_i| \leq 50000$ ), каждая пара с новой строки, числа в парах отделены друг от друга одним или несколькими пробелами.

### Формат выходных данных

В первой строке выведите число  $M$  — количество отрезков в искомом объединении. В следующих  $M$  строках выведите сами эти отрезки в том же формате, что и во входном файле. Список отрезков необходимо упорядочить по возрастанию левого конца.

### Примеры

merge.in	merge.out
4 0 2 4 5 1 3 5 6	2 0 3 4 6

## Задача F. Для любителей статистики

Имя входного файла: `queries.in`  
Имя выходного файла: `queries.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

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  $n$ ,  $0 < n < 70000$ . 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  $q$ ,  $0 < q < 70000$ . 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
5 123 666 314 666 434 5 1 5 314 1 5 578 2 4 666 4 4 713 1 1 123	10101

## Задача G. Вербочки

Имя входного файла: `ropes.in`  
Имя выходного файла: `ropes.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

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.

### 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

### Формат входных данных

В первой строке заданы два числа  $-N$  ( $0 \leq N \leq 10001$ ) и  $K$  ( $0 \leq K \leq 10001$ ). 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 число — максимальная длина веревочек, удовлетворяющих условию в сантиметрах. В случае, если лагерь закроют, выведите 0.

### Пример

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

## Задача Н. Точки и отрезки

Имя входного файла: `segments.in`  
Имя выходного файла: `segments.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

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 по два целых числа  $a_i$  и  $b_i$  — координаты концов соответствующего отрезка. В последней строке  $m$  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 количество отрезков, в которых она содержится.