

## Задача А. Цепная реакция

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Мало кто знает, почему магическая технология такая эффективная. На самом деле любой магический кристалл можно представить в виде  $n$  узлов и  $m$  двухсторонних связей между ними. Иногда в некоторых узлах вспыхивают искры энергии, которые затем распространяются в соседние узлы, из них — в соседние с ними, и так далее.

Недавно исследователи академии Пилтовера во главе с Хеймердингером выяснили, что делает магические кристаллы особенно сильными. Некоторые из связей между узлами являются *накопительными*. Это означает, что энергия может проходить через них только в определенные периоды времени, а вне этих периодов скапливается в концах этих связей и сохраняется в них.

Для каждой связи известно, сколько времени уходит на перемещение энергии из одного ее конца в другой. Также, для всех *накопительных* связей известно, что они пропускают энергию в одни и те же интервалы времени. Эти интервалы задаются списком пар моментов времени  $(x_i, y_i)$ , означающими, что между моментами времени  $x_i$  и  $y_i$  включительно каждая связь открыта и пропускает энергию. В остальные моменты времени они закрыты, и энергия «задерживается» в их концах.

**Важно отметить**, что если *накопительная* связь закрывается в тот момент, когда энергия перемещается по ней, энергия продолжает двигаться в ту сторону, в которую двигалась. Иными словами, если в момент  $t$  связь длиной  $w$  открыта, и в этот же момент  $t$  к одному ее концу приходит искра энергии, то в момент  $t + w$  она доберется до второго конца, даже если связь закроется к этому времени.

Для создания нового магического оружия необходимо определить, в какой момент времени энергия впервые дойдет до узла  $v$ , если искра зародится в узле  $u$  в момент времени  $t_0$ , или что энергия не дойдет до узла  $v$  вообще.

### Формат входных данных

В первой строке ввода даны три целых числа  $n, m, k$  — количество узлов кристалла и связей между ними, а также количество пар моментов времени, между которыми все накопительные связи открыты ( $2 \leq n \leq 10^5$ ;  $1 \leq m \leq 10^6$ ,  $1 \leq k \leq 10^5$ ).

В  $i$ -й из следующих  $m$  строк через пробел даны четыре целых числа  $a_i, b_i, w_i$  и  $f_i$ . Первые два числа  $a_i$  и  $b_i$  — номера узлов, которые соединяет  $i$ -я связь ( $1 \leq a_i, b_i \leq n$ ;  $a_i \neq b_i$ ). Число  $w_i$  — время, за которое энергия перемещается между концами связи ( $1 \leq w_i \leq 10^9$ ). Число  $f_i$  задает, является ли  $i$ -я связь *накопительной*. Он равен 1, если является, и 0 иначе.

В каждой из следующих  $k$  строк записаны по два целых числа  $x_i, y_i$  — моменты времени, между которыми (включительно) каждая накопительная связь открыта и может перемещать энергию ( $1 \leq x_i \leq y_i \leq 10^{18}$ ;  $y_{i-1} < x_i$  для всех  $i$ ).

В последней строке ввода даны три целых числа  $u, v$  и  $t_0$  — номера стартового и конечного узла, а также время зарождения искры энергии в стартовом узле ( $1 \leq u, v \leq n$ ;  $1 \leq t_0 \leq 10^{18}$ ).

### Формат выходных данных

Выведите одно целое число — через какое минимальное время энергия впервые дойдет от узла  $u$  до узла  $v$ , начав движение в момент времени  $t_0$ . Если она не дойдет до  $v$  никогда, выведите «-1» (без кавычек).

## Примеры

стандартный ввод	стандартный вывод
2 1 1 1 2 1 0 2 3 1 2 3	1
6 8 2 1 2 1 0 3 5 5 0 3 4 1 1 2 4 2 0 3 6 3 1 4 6 1 0 5 6 1 1 1 6 1 1 6 6 39 40 2 5 8	32

## Задача В. Производство Мерцания

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Чтобы синтезировать Мерцание, Силко необходимы специальные станки. Но перед тем, как их использовать, станки нужно раздобыть в городе и доставить на фабрику.

Город представляет собой взвешенный неориентированный граф из  $n$  вершин и  $m$  ребер, фабрика находится в вершине с номером 1. В некоторых вершинах находятся станки,  $i$ -й из которых характеризуется тройкой чисел  $(v_i, h_i, t_i)$ . Здесь  $v_i$  — номер вершины графа, в которой он находится,  $h_i$  — время, необходимое на установку и разогрев станка, и  $t_i$  — скорость производства компонентов Мерцания.

У Силко есть  $k$  свободных курьеров, каждый из которых может привезти один станок. Доставивший один станок курьер должен залечь на дно, и больше за станками отправиться не может. Таким образом, всего Силко может доставить на фабрику не более  $k$  станков.

Все курьеры начинают одновременно в момент времени 0, находясь на фабрике. Чтобы добраться до станка, курьеру нужно потратить время, равное сумме весов ребер на кратчайшем пути до вершины, в которой станок находится. Если станок находится в вершине 1, на его доставку не требуется время, однако в любом случае потребуется один курьер.

Когда станок прибывает на фабрику, сначала тратится  $h_i$  единиц времени на его подготовку, после чего он начинает непрерывно производить один компонент в  $t_i$  единиц времени. Какое минимальное время нужно для производства  $V$  компонентов?

### Формат входных данных

В первой строке ввода через пробел перечислены три целых числа  $n$ ,  $m$  и  $k$  — количество вершин графа, ребер графа и курьеров, соответственно ( $2 \leq n, m, k \leq 2 \cdot 10^5$ ).

В  $i$ -й из следующих  $m$  строк через пробел даны три целых числа  $a_i$ ,  $b_i$  и  $c_i$ , означающие, что между вершинами  $a_i$  и  $b_i$  проведено ребро веса  $c_i$  ( $1 \leq a_i, b_i \leq n$ ;  $a_i \neq b_i$ ;  $1 \leq c_i \leq 10^9$ ). Гарантируется, что все перечисленные ребра различны.

В следующей строке дано единственное целое число  $s$  — количество станков, расположенных в городе ( $1 \leq s \leq 2 \cdot 10^5$ ).

В  $i$ -й из следующих  $s$  строк дано описание  $i$ -го станка, состоящее из трех целых чисел  $v_i$ ,  $h_i$  и  $t_i$  — вершины, в которой расположен станок, времени его разогрева и времени производства станком одного компонента, соответственно ( $1 \leq v_i \leq n$ ;  $1 \leq h_i, t_i \leq 10^9$ ).

В последней строке дано единственное целое число  $V$  — количество компонентов, которые необходимо произвести ( $1 \leq V \leq 10^9$ ).

### Формат выходных данных

Выведите единственное целое число — ответ на задачу — минимальное время, необходимое для производства  $V$  компонентов.

## Примеры

стандартный ввод	стандартный вывод
3 3 2 1 2 5 1 3 2 3 2 2 3 1 15 1 2 1 1 3 3 2 10	15
4 4 4 4 2 5 1 3 3 3 4 16 2 1 9 10 3 18 8 1 2 12 4 8 19 2 9 15 2 12 2 4 20 4 3 11 14 2 5 3 4 19 1 1 1 20 3	26

## Задача С. Опасные игры

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Это интерактивная задача.

Однажды утром Вай получила посылку от Джинкс, где обнаружила радиопередатчик и схематический рисунок бомбы с подписью «НАЙДИ МЕНЯ». Передатчик, как оказалось, указывает направление до бомбы, но может сделать это ограниченное число раз, о чем многозначительно говорит все уменьшающаяся красная цифра на экране.

Вай знает, что бомба спрятана где-то в городе на участке размером  $n \times n$ . Она может встать в любую клетку участка и активировать передатчик, который сообщит, где относительно неё находится взрывчатка. Клетки задаются координатами, первая координата — номер строки сверху-вниз, вторая — номер столбца слева-направо. Если Вай уже в клетке с бомбой, то передатчик выдаст «OK», и бомба будет деактивирована, иначе выдаст направление, в котором находится бомба относительно текущей позиции Вай.

Направление может принимать одно из значений «N», «S», «W», «E», «NW», «NE», «SW» или «SE». 'N' означает вверх, 'S' — вниз, 'W' — влево, 'E' — вправо, а их комбинация, соответственно, — комбинацию из двух направлений. К примеру, если ответ будет «NE», то это значит, что клетка с бомбой находится сверху справа от текущей.

### Протокол взаимодействия

В первой строке интерактор сообщает вашей программе целое число  $n$  — длину стороны поля, на котором ведется поиск ( $1 \leq n \leq 10^9$ ).

Далее ваша программа должна следовать протоколу взаимодействия с интерактором. Весь протокол взаимодействия разбивается на запросы, которые совершает ваша программа, и ответы на них, которые дает интерактор.

Ваша программа должна передавать интерактору запросы в формате « $x\ y$ » (без кавычек), где  $x$  и  $y$  — координаты интересующей вас клетки по вертикали и горизонтали, соответственно ( $1 \leq x, y, \leq n$ ). Каждый запрос должен быть выведен на отдельной строке и должен заканчиваться сбросом буфера потока вывода (`cout.flush()` в C++, `System.out.flush()` в Java, `sys.stdout.flush()` в Python).

В ответ на запрос  $(x, y)$  интерактор сообщит относительное положение искомой клетки поля на отдельной строке. Если искомая клетка не совпадает с  $(x, y)$ , ответ интерактора будет являться конкатенацией двух частей, первая из которых может быть равна 'N', 'S' или пустой строке, а вторая — 'W', 'E' или пустой строке. Каждый символ отвечает за положение искомой клетки относительно  $(x, y)$ . Символ 'N' означает, что искомая клетка находится выше, 'S' — ниже, 'W' — левее, 'E' — правее.

Если запросом  $(x, y)$  ваша программа угадывает искомую клетку, интерактор отвечает в новой строке словом «OK» (без кавычек) и завершается. Считав такой ответ, ваша программа тоже должна завершиться, иначе может быть получен вердикт TL или IL.

Всего ваша программа может сделать не более 50 запросов к интерактору. Как только ваша программа превышает это ограничение, интерактор завершается с ошибкой WA. Если ваша программа при этом не завершается, может быть получен вердикт TL или IL.

## Примеры

стандартный ввод	стандартный вывод
3	2 2
NW	1 1
OK	
5	2 2
SE	3 5
SW	4 4
W	4 3
OK	

## Задача D. Тренировки миротворцев

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Миротворцам Пилтовера часто приходится подавлять волнения в Зауне, и отработка таких сценариев – неотъемлемая часть учений.

В данный момент стражи порядка работают в тройках и отрабатывают следующую формацию: Пилтовер представляется как координатная решетка, в нуле которой изначально находятся все трое. За ход один из миротворцев может сместиться на 1 в любом из четырех направлений: влево, вверх, вправо, вниз. Цель учений — как можно быстрее, то есть за минимальное число ходов, образовать треугольник, площадью хотя бы  $\frac{s}{2}$ .

Иными словами, требуется определить, какое минимальное число ходов необходимо сделать трем миротворцам, чтобы стоять в вершинах треугольника площади не меньше  $\frac{s}{2}$ .

### Формат входных данных

В единственной строке ввода дано целое число  $s$  — удвоенная желаемая площадь треугольника, ограниченного миротворцами ( $0 \leq s \leq 10^{18}$ ).

### Формат выходных данных

Выведите единственное целое число  $c$  — минимальное суммарное количество перемещений, которые необходимо совершить, чтобы получить такой треугольник.

### Примеры

стандартный ввод	стандартный вывод
1	2
3	4

## Задача Е. Джинкс и лагерь миротворцев

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Джинкс планирует нападение на лагерь миротворцев.

База миротворцев представляется, как поле на координатной плоскости.

Героиня знает, что на страже лагеря  $n$  миротворцев. Каждый из них имеет силу  $v_i$ . Каждый страж отвечает за некоторый прямоугольник на координатной плоскости. Каждый прямоугольник задается координатами левой верхней и правой нижней точки, то есть парой точек  $(x_1, y_1)$ ,  $(x_2, y_2)$  ( $x_1 \leq x_2$  и  $y_1 \leq y_2$ ). Будем считать, что точка  $(x, y)$  защищается стражем, отвечающим за прямоугольник  $(x_1, y_1)$ ,  $(x_2, y_2)$ , если  $x_1 \leq x \leq x_2$  и  $y_1 \leq y \leq y_2$ .

Одна точка базы может защищаться несколькими стражами, то есть прямоугольниками за которые они отвечают могут пересекаться. К сожалению (или к счастью для Джинкс), от этого мало пользы, так как защищенность точки  $(x, y)$  на базе  $s_{x,y}$  равна минимальной силе среди всех миротворцев, отвечающих за данную точку, или  $-1$ , если ни один страж не защищает эту точку.

К базе миротворцев существует  $m$  подходов.  $j$ -й подход ведет или ко всем точкам внутри прямоугольника, заданного крайними точками  $(x, y_1)$  и  $(x, y_2)$ , или прямоугольника, заданного крайними точками  $(x_1, y)$  и  $(x_2, y)$ . То есть выбрав некоторый подход, может попасть в точку  $(\bar{x}, \bar{y})$  базы миротворцев, если (в зависимости от вида подхода)  $\bar{x} = x$  и  $y_1 \leq y \leq y_2$  или  $\bar{y} = y$  и  $x_1 \leq x \leq x_2$ .

Для каждого из подходов Джинкс хочет узнать минимальную защищенности среди всех точек, в которые она может попасть через этот подход, то есть найти  $\min_{\bar{x}, \bar{y}} s_{\bar{x}, \bar{y}}$ , для всех  $(\bar{x}, \bar{y})$ , если (в зависимости от вида подхода)  $\bar{x} = x$  и  $y_1 \leq y \leq y_2$  или  $\bar{y} = y$  и  $x_1 \leq x \leq x_2$ .

### Формат входных данных

В первой строке находятся два целых числа  $n$  и  $m$  ( $1 \leq n \leq 3 \cdot 10^4$ ;  $1 \leq m \leq 10^5$ ) — количество стражей и количество подходов к базе.

В следующих  $n$  строках следует описание охранников базы. Строка номер  $i$  содержит пять целых чисел  $x_{i,1}$ ,  $x_{i,2}$ ,  $y_{i,1}$ ,  $y_{i,2}$  и  $v_i$  —  $x$ -координата верхней левой точки,  $x$ -координата нижней правой точки,  $y$ -координата верхней левой точки,  $y$ -координата нижней правой точки  $i$ -го прямоугольника, за который отвечает  $i$ -й миротворец, а также его сила ( $1 \leq x_{i,1}, x_{i,2}, y_{i,1}, y_{i,2} \leq 2 \cdot 10^5$ ;  $0 \leq v_i \leq 10^9$ ).

В следующих  $m$  строках следует описание подходов.

Первый символ описания подхода равен либо 'x', либо 'y'. Если вида подхода равен 'x', то этот подход имеет вид  $(x, y_1)$  и  $(x, y_2)$ , и далее через пробел даны координаты прямоугольника  $x_j$ ,  $y_{j,1}$  и  $y_{j,2}$ . Иначе — это подход типа  $(x_1, y)$  и  $(x_2, y)$ , и далее идут координаты прямоугольника  $y_j$ ,  $x_{j,1}$  и  $x_{j,2}$ .

### Формат выходных данных

Для каждого из  $m$  подходов выведите минимальную защищенности среди всех точек, в которые она может попасть через этот подход в отдельной строке.

### Пример

стандартный ввод	стандартный вывод
4 4	1
1 4 1 4 3	2
1 3 2 4 2	-1
1 5 2 4 4	3
1 2 1 3 1	
x 2 3 5	
x 3 1 4	
x 5 1 1	
x 4 1 3	

## Задача F. Поезда в Зауне

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Давайте на минуту представим, что было бы, если бы Заун получил свою независимость от Пилтовера. Как для около-индустриального квартала, в первую очередь для Зауна было бы важно наладить быстрое транспортное сообщение между различными производственными точками.

Был создан план, в рамках которого в Зауне последовательно строятся  $n$  железнодорожных линий. Сначала будет открыта первая линия, затем, спустя время, вторая, и так далее. Линии могут пересекаться (возможно, не один раз). Если линии  $i$  и  $j$  пересекаются, то в этом месте транспорт может переезжать с  $i$ -й ветки на  $j$ -ю или наоборот.

Также известно, что в течение дня по  $i$ -й линии (если она уже построена) будут курсировать ровно  $a_i$  поездов. А по ночам жизнь в квартале замирает, и все поезда должны находиться в депо (не обязательно на своей ветке). В депо вместимости  $A$  могут находиться не более  $A$  поездов. Каждый поезд может быть размещен в любом достижимом депо (таким образом, если в депо на линии  $j$  есть место, и ветки  $i$  и  $j$  пересекаются, то поезд, курсирующий днем по ветке  $i$ , может ночью расположиться в депо на ветке  $j$ ).

Можно построить произвольное количество депо любой вместимости на любых ветках еще до их официального открытия. Ваша цель — чтобы в любой момент времени все поезда на открытых ветках можно было разместить в достижимых депо. А поскольку постройка депо очень трудозатратна, хочется минимизировать их суммарную вместимость, а при равенстве — минимизировать их количество.

Найдите  $k$  — минимальное количество депо минимальной суммарной вместимости, которое необходимо построить, чтобы открывать линии в запланированном порядке. Про каждое из  $k$  депо укажите на какой из линий оно должно быть построено и какую вместимость должно иметь. Обратите внимание, что следует минимизировать суммарную вместимость, а среди таких вариантов уже выбрать с минимальным количеством.

### Формат входных данных

В первой строке ввода дано целое число  $n$  — количество линий, которое планируется построить ( $1 \leq n \leq 10^6$ ).

В каждой из следующих  $2 \cdot n$  строк содержится описание запросов в формате, описанном ниже.

В первой строке описания  $i$ -й линии записано единственное целое число  $a_i$  — количество поездов на этой линии ( $1 \leq a_i \leq 10^6$ ). Во второй строке описания запроса сначала дано целое число  $c_i$  — количество уже построенных линий, пересекаемых данной, ( $0 \leq c_i \leq 2 \cdot 10^6$ ), а затем через пробел перечислены  $c_i$  целых чисел — номера линий, с которыми пересекается новая. Гарантируется, что для всех перечисленных  $t$  выполняется  $1 \leq t < i$ .

Гарантируется, что сумма  $c_i$  по всем  $i$  не превосходит  $2 \cdot 10^6$ .

### Формат выходных данных

В первой строке выведите единственное целое число  $k$  — количество депо, которые вы собираетесь построить.

В  $i$ -й из следующих  $k$  строк выведите через пробел по два числа: номер линии (от 1 до  $n$ ) и вместимость  $i$ -го депо.

## Пример

стандартный ввод	стандартный вывод
7	4
20	1 20
0	2 10
10	3 120
0	4 50
30	
0	
40	
0	
10	
2 1 4	
40	
2 2 3	
50	
1 2	

## Задача G. Взрывоопасная лестница

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Джинкс совершила очередное ограбление, и теперь пытается убежать от преследователей. На ее пути отступления находится лестница, состоящая из строительных блоков Хекстека. После прохода по ней, она собирается эту лестницу взорвать.

Лестница состоит из  $n$  уровней, каждый из уровней состоит из какого-то количества блоков. А именно,  $i$ -й сверху уровень состоит из  $i$  строительных блоков. Каждый блок Хекстека обладает какой-то магической силой. Так как сила любой лестницы в ее фундаменте, то магическая сила взрыва лестницы зависит только от самого нижнего ее уровня. А именно, чем *лексикографически меньше* последовательность магических блоков последнего уровня в лестнице, тем сильнее будет взрыв.

Напомним, что последовательность  $a$  *лексикографически меньше* последовательности  $b$ , если существует такой индекс  $i$ , что  $a_1 = b_1, a_2 = b_2, \dots, a_{i-1} = b_{i-1}$ , а  $a_i < b_i$ . Иными словами, что в первой различающейся позиции, в  $a$  стоит меньшее значение.

У Джинкс есть в запасе немного времени, поэтому она может переставить уровни в любом порядке, после чего блоки, под которыми появилось пустое место, падают вниз под действием гравитации. Времени не так много, как хотелось бы, поэтому такую операцию она сможет осуществить лишь один раз. Теперь ей интересно, в каком порядке нужно переставить уровни, чтобы сила взрыва была максимальна.

### Формат входных данных

В первой строке находится единственное число  $n$  — количество уровней лестницы ( $1 \leq n \leq 1000$ ).

В  $i$ -й из следующих  $n$  строк через пробел перечислены  $i$  чисел  $a_{i,1}, \dots, a_{i,i}$  — магические силы блоков в  $i$ -м ряду ( $0 \leq a_{i,j} \leq 10^9$ ).

### Формат выходных данных

Выведите перестановку из  $n$  чисел от 1 до  $n$  — порядок, в котором Джинкс должна расположить уровни сверху вниз, чтобы сила взрыва была максимальна.

### Пример

стандартный ввод	стандартный вывод
3	3 2 1
1	
2 2	
1 3 3	

### Замечание

В приведенном примере, изначально лестница выглядит так:

```
1
2 2
1 3 3
```

После смены порядка уровней:

```
1 3 3
2 2
1
```

После действия гравитации:

```
1
2 3
1 2 3
```

## Задача Н. Компонентная химия

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Силко вместе со своим главным ученым Сингедом продолжает создавать в лаборатории опасные препараты.

Вещество, над которым сейчас занимаются герои должно называться «Мерцание». Они планируют, что вещество будет иметь эффективность  $e$  и состоять из простых химических компонентов. При этом, может быть такое, что некоторый компонент входит в состав несколько раз. Все химические компоненты так же имеют некоторую эффективность.

Ученые еще не определились с конечным составом препарата, однако точно знают, что его сила будет равна количеству компонент  $i$  и  $j$  ( $i \neq j$ ), таких, что сумма их эффективностей, то есть  $a_i + a_j$ , кратна  $e$ .

Герои хотят получить препарат с оптимальной силой, поэтому они экспериментируют с составом. Изначально в составе препарата не было компонентов. Вам требуется обработать  $m$  запросов вида:

1. Добавить химический компонент с эффективностью  $a_i$  в состав;
2. Убрать химический компонент  $a_i$  из состава препарата. Если в составе препарата несколько компонент с такой эффективностью, то убрать только один компонент с такой эффективностью.

После каждой из операций найдите силу препарата с текущим составом.

### Формат входных данных

В первой строке входных через пробел даны два целых числа  $n$  и  $m$  — количество запросов и планируемая эффективность препарата ( $1 \leq n \leq 10^5$ ;  $2 \leq m \leq 10^9$ ).

В следующих  $m$  строках даются пары чисел  $t_i$  и  $a_i$  — тип операции (1, если необходимо добавить препарат с данной эффективностью, иначе 2) и эффективность добавляемого компонента ( $t_i \in \{1, 2\}$ ;  $-10^9 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите  $n$  строк, каждая из которых содержит силу препарата после применения первых  $i$  запросов.

### Пример

стандартный ввод	стандартный вывод
7 19	0
1 10	0
1 13	1
1 25	2
1 6	1
2 25	2
1 9	1
2 10	

## Задача I. Палиндромная шифровка

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Кейтлин Кирамман удалось поймать одного из приспешников Силко и найти у него зашифрованное сообщение из  $n$  строк  $s_1, s_2, \dots, s_n$ , составленных из маленьких латинских букв. Длина каждой строки  $|s_i|$  не превосходит 1000.

К сожалению, сообщение оказалось неполным, и, не имея всего текста, расшифровать послание нельзя. Известно, что каждая полученная строка  $s_i$  является правой долей  $i$ -й части сообщения, то есть у каждой строки не хватает некоторого префикса (возможно, пустого). Также известно, что изначально каждая часть сообщения была палиндромом. То есть, в конечном итоге, каждая  $s_i$  — это суффикс некоторого палиндрома.

В поисках недостающих частей Кейтлин обратилась в архив, где ей выдали  $m$  строк  $t_1, t_2, \dots, t_m$ , которые потенциально могут дополнять строки  $s_i$  до палиндромов. Каждая строка  $t_i$  также состоит из маленьких латинских букв.

Кейтлин хочет проверить, могут ли полученные в архиве материалы быть кусками исходной шифровки. Для этого ей для каждой строки  $t_i$  нужно понять, существует ли такое  $j$ , что  $t_i + s_j$  — палиндром (здесь за знак сложения обозначена операция конкатенации).

### Формат входных данных

В первой строке ввода через пробел даны два целых числа  $n$  и  $m$  ( $1 \leq n \leq 1000$ ;  $1 \leq m \leq 10^6$ ).

Во второй строке перечислены  $n$  строк  $s_1, s_2, \dots, s_n$ , разделенные пробелами.

Третья строка в том же формате содержит  $m$  строк  $t_1, t_2, \dots, t_m$ , разделенные пробелами. Гарантируется, что  $\sum_{i=1}^m |t_i| \leq 10^6$ .

### Формат выходных данных

Для каждой строки  $t_i$  выведите на отдельной  $i$ -й строке слово «YES» (без кавычек), если существует такое  $j$ , что  $t_i + s_j$  — палиндром, и «NO» иначе.

### Пример

стандартный ввод	стандартный вывод
3 4	NO
aba mogus oba	YES
ba abac ab aaa	YES
	NO

## Задача J. Магические часы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Ученые из Пилтовера заполучили очень загадочные магические часы. Странный ход стрелок натолкнул их на мысль, что в этих часах могут быть зашифрованы важные сведения.

Часы состоят из циферблата на  $60 \cdot 12 \cdot t$  делений, пронумерованных от 0 до  $720t - 1$  по часовой стрелке включительно, а также, собственно, из минутной и часовой стрелок. Каждую  $\frac{1}{t}$ -ю минуты (назовем этот интервал времени *тиком*) минутная стрелка проходит 12 делений, а часовая — 1. Таким образом, на обычных часах с таким устройством за один час, то есть за 60 минут, минутная стрелка проходит весь круг, а часовая —  $\frac{1}{12}$ -ю круга.

Странность часов заключается в том, что когда на очередном тике минутная стрелка должна обогнать или догнать часовую, она телепортируется в начало. Таким образом, если минутная стрелка указывает на деление номер  $m$ , а часовая — на  $h$ , и расстояние между ними равно  $d = (h - m) \bmod (720t)$  делений, то если  $0 < d < 12$ , после следующего *тика* минутная стрелка окажется на нулевом делении (тогда как часовая спокойно продолжит свой ход).

Хеймердингер выдвинул  $q$  теорий относительно природы и способностей таких часов, и для проверки  $i$ -й теории нужно научиться отвечать, через сколько времени стрелки из состояния  $s_{i,1}$  перейдут в состояние  $s_{i,2}$ .

### Формат входных данных

В первой строке ввода через пробел даны два целых числа  $t$  и  $q$  —  $\frac{1}{720}$ -я количества делений на часах и количество запросов ( $1 \leq t \leq 1500$ ;  $1 \leq q \leq 10^6$ ).

В  $i$ -й из следующих  $q$  строк дано описание  $i$ -го запроса, состоящее из четырех целых чисел  $h_1$ ,  $m_1$ ,  $h_2$  и  $m_2$ , разделенных пробелами — номеров делений, на которые указывают часовая и минутная стрелка в начальном и конечном состояниях, соответственно ( $0 \leq h_{1,2}, m_{1,2} < 720t$ ).

### Формат выходных данных

Выведите  $q$  строк — ответы на все запросы Хеймердингера, каждый в своей строке.

В качестве ответа на запрос выведите минимальное число *тиков*, спустя которое часы перейдут из состояния  $(h_1, m_1)$  в состояние  $(h_2, m_2)$ , либо «-1» (без кавычек), если часы никогда не придут во второе состояния, стартуя из указанного начального.

### Примеры

стандартный ввод	стандартный вывод
1 4	1
0 0 1 12	60
0 0 60 0	1
11 0 12 0	-1
12 0 13 0	
2 5	827
1201 1317 588 396	-1
658 1196 102 84	-1
442 8 1246 1200	660
79 0 739 0	5503
355 286 98 72	