

## Problem B. Сообщение пришельцам

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

В известном в Байтландии научно-популярном журнале вышла статья о попытках контакта с внеземными цивилизациями.

В статье рассказывается, что по инициативе Байтландского Общества Контакттов с внеземными цивилизациями на берегу моря неподалеку от Байтсбурга был выложен узор из белых и чёрных плит, расположенных последовательно друг за другом. В узоре должно было быть зашифровано сообщение. Но где-то на этапе между утверждением проекта и строительством произошло непонимание, так что плиты оказались установлены не в том порядке, и смысл сообщения поменялся.

Для перестановки плит используется специальный кран с двумя захватами. Алгоритм перестановки выглядит следующим образом:

- переместить кран так, чтобы он стоял над первой панелью, прицепить первый захват к панели;
- переместить кран так, чтобы он стоял над второй панелью, прицепить второй захват к панели;
- нажать кнопку «Обмен»; панели поменяются местами, при этом кран останется на месте (то есть над второй панелью).

Каждое перемещение крана между двумя соседними плитами обходится в 1 байтландский тугрик; процедура обмена не оплачивается.

Требуется выбрать начальное положение крана так, чтобы превратить начальную расстановку плит в конечную, затратив как можно меньшую сумму.

### Input

Вход состоит из нескольких тестовых примеров.

Каждый тестовый пример состоит из трёх строк. Первая строка тестового примера содержит целое число  $N$  — количество пластин ( $2 \leq N \leq 16$ ). Вторая строка содержит начальное положение плит, третья строка — конечное положение плит. Каждая строка состоит из  $N$  символов 'B' и 'W', обозначающих чёрные и белые плиты соответственно. Плиты одного цвета неразличимы. Гарантируется, что существует хотя бы одна последовательность команд, которая переводит начальное положение в конечное.

Входные данные заканчиваются тестовым примером с  $N = 0$ , обрабатывать который не требуется.

### Output

Для каждого тестового примера выведите одно число — минимальную стоимость перестановки плит.

### Example

standard input	standard output
4	3
BWBW	9
WBWB	
8	
BBBBWBWW	
BBWWBWB	
0	

## Problem C. Каббалистический квадрат

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 256 mebibytes

Чтобы вычислить *цифровой корень* числа, складывают цифры в его десятичной записи. Если результат больше 9, у полученного числа также складываются все цифры и так далее пока не получится однозначное число.

Например, посчитаем цифровой корень числа 123454:

123454:  $1 + 2 + 3 + 4 + 5 + 4 = 19$

19:  $1 + 9 = 10$

10:  $1 + 0 = 1$

Тем самым числовой корень числа 123454 равен 1:  $DR(123454) = 1$ .

Вам дан массив  $a_i$ , состоящий из 9 строго возрастающих целых положительных чисел, не превосходящих 1000. *Каббалистическим квадратом* этих чисел называется таблица  $V$  размером  $9 \times 9$  cells, в которой клетка  $V_{i,j}$  содержит цифровой корень произведения  $a_i \cdot a_j$  ( $V_{i,j} = DR(a_i \cdot a_j)$ ).

Ваша задача — написать программу, которая, используя заданную таблицу  $V$ , реконструирует числа  $a_1, a_2, \dots, a_9$  так, что  $V_{i,j} = DR(a_i \cdot a_j)$  или определяет, что это невозможно.

### Input

Входные данные состоят из 9 строк, каждая из которых содержит по 9 целых чисел  $V_{i,j}$  ( $1 \leq V_{i,j} \leq 9$ ).  $i$ -я из этих строк должна соответствовать цифровому корню  $a_i \cdot a_1, a_i \cdot a_2, \dots, a_i \cdot a_9$

### Output

Если решения не существует, выведите "No solution". В противном случае выведите 9 целых положительных чисел  $a_1, a_2, \dots, a_9$  ( $a_i < a_{i+1}$  для  $1 \leq i \leq 8, 1 \leq a_i \leq 1000$ ). Если решений несколько, выведите любое из них.

### Examples

standard input	standard output
1 2 3 4 5 6 7 8 9 2 4 6 8 1 3 5 7 9 3 6 9 3 6 9 3 6 9 4 8 3 7 2 6 1 5 9 5 1 6 2 7 3 8 4 9 6 3 9 6 3 9 6 3 9 7 5 3 1 8 6 4 2 9 8 7 6 5 4 3 2 1 9 9 9 9 9 9 9 9 9 9	1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 1 2 4 6 8 1 3 5 7 9 3 6 9 3 6 9 3 6 9 4 8 3 7 2 6 1 5 9 5 1 6 2 7 3 8 4 9 6 3 9 6 3 9 6 3 9 7 5 3 1 8 6 4 2 9 8 7 6 5 4 3 2 1 9 9 9 9 9 9 9 9 9 9	No solution

## Problem D. Tower Defence

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 256 mebibytes

В игре «Tower Defence» ваша задача — строить башни для защиты своего королевства от монстров. Перед атакой очередной группы монстров Вы хотите узнать, сможете ли Вы ей противостоять.

Монстры движутся по прямой. Прямая разбита на  $N$  зон защиты, занумерованных последовательными целыми положительными числами от 1 до  $N$ . Перед наступлением вы построили  $M$  башен. Для каждой башни задан диапазон обстрела  $[L, R]$ , обозначающий, что башня атакует монстров в зонах защиты  $i$ , где  $L \leq i \leq R$ .

Как только монстр попадает в зону, атакуемую одной или несколькими башнями, каждая из этих башен производит один выстрел в монстра. Например, башня с диапазоном обстрела  $[1, 3]$  будет атаковать монстра трижды — в зоне 1, в зоне 2 и в зоне 3.

$i$ -й монстр появляется в зоне  $X_i$  и сразу же начинает двигаться в направлении зоны  $N$ . Также Вам известно количество здоровья  $H_i$  для каждого монстра и ущерб  $D_i$ , наносимый  $i$ -й башней. При атаке монстра  $i$ -й башней его здоровье уменьшается на  $D_i$ .

Если количество здоровья монстра упало до нуля или менее, монстр погибает и исчезает с карты.

Ваша задача — подсчитать количество монстров, которые пройдут за  $N$ -ю зону.

### Input

Первая строка входа содержит целое число  $N$  ( $0 < N \leq 10^5$ ) — количество зон защиты. Вторая строка содержит целое число  $M$  ( $0 < M \leq 10^5$ ) — количество башен.  $i$ -я из последующих  $M$  строк содержит по три целых числа  $L_i, R_i, D_i$  ( $1 \leq L_i \leq R_i \leq N$ ,  $0 < D_i \leq 1000$ ), задающих диапазон обстрела  $[L, R]$  и ущерб  $D$  для  $i$ -й башни.

В следующей строке задано одно целое число  $K$  ( $0 < K \leq 10^5$ ) — количество атакующих монстров. Каждая из последующих  $K$  строк описывает одного монстра и содержит два целых числа  $H_i$  и  $X_i$  ( $0 < H_i \leq 10^{18}$ ,  $1 \leq X_i \leq N$ ), обозначающих количество здоровья у монстра и блок, в котором монстр появляется.

### Output

Выведите одну строку — количество выживших монстров.

### Example

standard input	standard output
5 2 1 3 1 5 5 2 5 1 3 3 1 5 2 7 3 9 1	3

## Problem E. Подводный клад

Input file: *standard input*  
Output file: *standard output*  
Time limit: 7 seconds  
Memory limit: 256 mebibytes

В подводной пещере спрятано несколько золотых слитков.

Пещера состоит из нескольких камер, соединённых туннелями. Камера с номером 0 находится у поверхности воды. Для каждого коридора известно количество кислорода, которое требуется для того, чтобы проплыть этот коридор. Количество кислорода в баллоне акваланга лимитировано.

В распоряжении экспедиции есть карта пещеры и список камер, в которых есть золотые слитки... но нет запасных баллонов с кислородом к аквалангу. Требуется построить маршрут аквалангиста так, чтобы он стартовал в камере с номером 0 и вернулся назад, собрав как можно больше слитков и не выйдя за лимит потреблённого кислорода.

### Input

Первая строка входных данных содержит одно целое число  $t$  ( $0 < t \leq 2000$ ) — количество тестовых примеров в файле. Первая строка каждого тестового примера содержит два целых числа  $n$  и  $m$  — количество камер и количество туннелей, их соединяющих ( $1 \leq n \leq 10^4$   $0 \leq m \leq 5 \cdot 10^4$ ).

Далее следуют  $m$  строк, задающих туннели. Каждая из этих строк содержит по три целых числа  $a$ ,  $b$  и  $l$  — номера камер, соединённых туннелем и количество воздуха, требуемое для того, чтобы преодолеть этот туннель ( $0 \leq a, b < n$ ;  $0 \leq l \leq 500$ ).

Далее следует строка, содержащая целое число  $i$  — количество слитков в пещере ( $0 \leq i \leq 8$ ).

За этой строкой следует строка, содержащая  $i$  целых чисел  $p_1, \dots, p_i$  — номера камер, в которых размещены слитки ( $0 \leq p_1, \dots, p_i < n$ ).

Завершает тестовый пример строка с одним целым числом  $V$  — объёмом потреблённого кислорода. ( $0 \leq V \leq 10^6$ ).

Суммарный объём входного файла не превосходит 16 MiB.

### Output

Для каждого тестового примера выведите одно число — максимальное количество слитков, которое сможет поднять аквалангист.

## Example

standard input	standard output
3	1
5 3	2
0 1 10	3
0 2 20	
0 3 30	
4	
1 2 3 4	
30	
5 3	
0 1 10	
0 2 20	
0 3 30	
4	
1 2 3 4	
60	
5 3	
0 1 10	
0 2 20	
0 3 30	
4	
1 2 3 4	
10000	

## Problem F. Профилактические работы

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 256 mebibytes

В одном из первых дата-центров компании Byndex  $N \times M$  серверов, расставленных в виде прямоугольника  $N \times M$ . К моменту начала профилактических работ все серверы включены. Во время профилактических работ системный администратор может за одну команду изменять статус всех серверов в блоках  $1 \times a$  и  $b \times 1$  (то есть включать выключенные сервера и выключать включённые). Два состояния дата-центра считаются различными, если хотя бы один сервер включён в первом состоянии дата-центра и выключен во втором и наоборот.

Определите число различных состояний дата-центра, возможных после  $K$  команд от системного администратора.

### Input

Первая строка входа содержит количество тестовых примеров  $T$  ( $1 \leq T \leq 600$ ). Далее перечислены тестовые примеры. Каждый тестовый пример содержит три целых числа  $N$ ,  $M$  и  $K$  ( $1 \leq N \leq 4$ ,  $1 \leq M \leq 10$ ,  $0 \leq K \leq NM$ ).

### Output

Для каждого тестового примера выведите в отдельной строке одно число — остаток от деления ответа к данному тестовому примеру на  $10^9 + 7$ .

### Example

standard input	standard output
2	16
2 2 2	512
3 3 4	

## Problem G. Призовые гонки

Input file: *standard input*  
Output file: *standard output*  
Time limit: 3 seconds  
Memory limit: 256 mebibytes

Одной из дисциплин Байтландских робоолимпийских игр являются призовые гонки. Гонки проходят на трассе, состоящей из  $N$  площадок, соединённых  $N - 1$  коридорами с двусторонним движением так, что между любыми двумя площадками можно проехать по цепочке коридоров.

На площадках можно найти сейфы и ключи. Для каждого сейфа известно, какой ключ его открывает и какая сумма бонусов (возможно, отрицательная) в нём содержится. Если робот оказывается на площадке, на которой лежат ключи, робот забирает все ключи; затем, если на площадке есть какие-то сейфы, к которым у робота есть ключ, он открывает все такие сейфы и количество набранных бонусов изменяется на сумму, которая была в сейфах. На старте количество набранных роботом бонусов равно нулю.

Точки старта и финиша выбираются разработчиком робота; между этими двумя точками робот должен следовать по кратчайшему пути. Какую наибольшую сумму бонусов можно собрать при оптимальном выборе точек?

### Input

Первая строка входных данных содержит одно целое число  $T$  ( $1 \leq T \leq 30$ ).

Каждый тестовый пример начинается строкой, содержащей два целых числа  $N$  и  $M$  ( $1 \leq n, m \leq 10^5$ ) — количество площадок и количество сейфов. Каждая из последующих  $n - 1$  строк содержит два целых числа  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq N$ ) — номера площадок, соединённых очередным коридором. Гарантируется, что расположение площадок и коридоров соответствует условию задачи. Каждая из последующих  $m$  строк содержит по три целых числа  $k_i$ ,  $s_i$  и  $b_i$  — номер площадки, на которой находится ключ от соответствующего сейфа, номер площадки, на которой находится сам сейф и количество бонусов, содержащееся в сейфе ( $1 \leq k_i, s_i \leq n$ ,  $-1000 \leq b_i \leq 1000$ ).

Гарантируется, что сумма  $N$  во всех тестовых примерах не превосходит  $6 \cdot 10^5$ . Гарантируется, что сумма  $M$  во всех тестовых примерах не превосходит  $6 \cdot 10^5$ .

### Output

Для каждого тестового примера выведите одно целое число — максимальную сумму бонусов, которую может собрать робот при правильном выборе точек старта и финиша.

### Example

standard input	standard output
2	123
4 2	9
2 1	
3 2	
4 3	
1 1 123	
2 4 -7	
4 3	
2 1	
3 1	
4 1	
2 1 2	
1 3 3	
1 3 4	