# Problem A. Линейные уравнения

| | |
|---|---|
| Input file: | `linear.in` |
| Output file: | `linear.out` |
| Time limit: | 2 секунды |
| Memory limit: | 256 мебибайт |

Система линейных уравнений, как всем известно, есть множество уравнений

$$\begin{aligned} a_{11}x_1 + \ldots + a_{1n}x_n &= b_1 \\ &\ldots \\ a_{n1}x_1 + \ldots + a_{nn}x_n &= b_n \end{aligned}$$

Ваша задача — решить её.

## Input

В первой строке входного файла записано целое число $n$ ($1 \leqslant n \leqslant 20$). В следующих $n$ строках записано по $n + 1$ целых чисел: $a_{i1}, \ldots, a_{in}, b_i$. Все эти числа не превышают 100 по абсолютному значению.

## Output

Первая строка выходного файла должна содержать одно из следующих сообщений:

- `impossible` — решений нет
- `infinity` — бесконечно много решений
- `single` — единственное решение. В этом случае вторая строка должна содержать $n$ чисел $x_1, \ldots, x_n$, разделенных пробелами. Решение должно быть выведено ровно с тремя знаками после десятичной точки.

## Examples

| linear.in | linear.out |
|---|---|
| 2<br>1 1 1<br>2 2 2 | infinity |
| 2<br>1 2 0<br>1 2 1 | impossible |
| 2<br>1 2 1<br>2 1 0 | single<br>-0.333 0.667 |

# Problem B. Векторы

| | |
|---|---|
| Input file: | vectors.in |
| Output file: | vectors.out |
| Time limit: | 2 секунды |
| Memory limit: | 256 мегабайта |

Задано $m$ векторов из нулей и единиц, длина каждого вектора равна $n$. Сложение векторов осуществляется покомпонентно по модулю 2.

Требуется для каждого вектора определить, можно ли его получить, сложив некоторое подмножество предыдущих, а для последнего вектора определить, какие вектора следует сложить, чтобы его получить.

## Input

Первая строка входного файла содержит два целых числа: $n$ и $m$ — длина векторов и количество векторов ($1 \le n, m \le 2000$). Следующие $m$ строк содержат по $n$ целых чисел, каждое из которых равно 0 или 1.

## Output

Для каждого вектора выведите "Yes" или "No" на отдельной строке. Если ответ для последнего вектора "Yes", выведите $m - 1$ число, каждое из которых должно быть равно 0 или 1. Выведите 1 для тех векторов, которые следует сложить, чтобы получить последний вектор.

## Example

| vectors.in | vectors.out |
|---|---|
| 4 6<br>0 0 0 0<br>1 0 0 1<br>1 0 1 0<br>0 0 1 1<br>0 1 0 0<br>0 1 1 1 | Yes<br>No<br>No<br>Yes<br>No<br>Yes<br>0 1 1 0 1 |

# Problem C. Связность графа

| | |
|---|---|
| Input file: | `disconnected.in` |
| Output file: | `disconnected.out` |
| Time limit: | 2 секунды |
| Memory limit: | 256 мебибайт |

Дан связный неориентированный граф. Вам поступают запросы вида: проверить, останется ли граф связным после удаления некоторого маленького множества ребер.

## Input

Первая строка входного файла содержит два числа — $N$ и $M$ ($1 \le N \le 10\,000$, $1 \le M \le 100\,000$), обозначающие число вершин и число ребер, соответственно. Следующие $M$ строк содержат описания ребер. Каждая строка состоит из двух чисел $a$ и $b$ — номера вершин, соединяемых соответствующим ребром. В графе нет петель и кратных ребер. Вершины графа нумеруются с единицы. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Следующая строка содержит единственное число $K$ ($1 \le K \le 100\,000$), обозначающие число запросов. Следующие $K$ строк содержат описания запросов. Каждое описание начинается с числа $C$ ($1 \le C \le 4$), обозначающее число ребер в запросе, далее следуют $C$ чисел, обозначающих номера ребер, входящих в запрос. Все ребра, входящие в запрос, являются различными.

## Output

Для каждого запроса выведите единственную строку. В $i$-ой строке должно содержаться слово «Connected», если удаление всех ребер из соответствующего сохранит связность графа, и «Disconnected» в противном случае.

## Examples

| disconnected.in | disconnected.out |
|---|---|
| 4 5<br>1 2<br>2 3<br>3 4<br>4 1<br>2 4<br>3<br>1 5<br>2 2 3<br>2 1 2 | Connected<br>Disconnected<br>Connected |

# Problem D. Звезда смерти-2

| | |
|---|---|
| Input file: | star2.in |
| Output file: | star2.out |
| Time limit: | 2 секунды |
| Memory limit: | 256 мебибайт |

Давным-давно в далекой-предалекой галактике...

Боевая космическая станция «Звезда Смерти» была спроектирована еще до начала Клонических войн. Через много лет она была передана в руки Империи для контроля над Внешними Территориями. «Звезда Смерти» имела более 100 миль в диаметре, была оснащена гравитонной пушкой, способной уничтожать целые планеты, а также могла нести на борту несколько тысяч истребителей. «Звезда Смерти» должна была наводить ужас на население и полностью исключить всякую возможность сопротивления властям Империи.

После того, как первая «Звезда Смерти» была уничтожена повстанцами, началось создание новой модели, еще более смертоносной. Новая модель, как и первая, имеет форму шара и может поступательно перемещаться в $N$-мерном пространстве. Она оснащена $M$ жестко закрепленными криптоновыми двигателями. Если на $i$-й двигатель подать $X$ единиц энергии, то вклад этого двигателя в $j$-ю координату вектора тяги составит $A_{ij} \cdot X$. В зависимости от режима работы, криптоновый двигатель может перемещать станцию вперед либо назад (в этом случае $X$ отрицательно). Итоговый вектор тяги равняется сумме вкладов каждого из $M$ двигателей.

Перед началом движения станции специальный навигационный модуль определяет необходимые координаты вектора тяги $(b_1, b_2, \ldots, b_N)$. Ваша программа должна вычислить количество энергии, которое необходимо подать на каждый из двигателей так, чтобы длина вектора разности суммарной тяги и необходимой тяги была минимальна. В случае, если ответ неоднозначен, требуется дополнительно минимизировать сумму квадратов величин энергии, подаваемой на двигатели.

## Input

В первой строке через пробел записаны числа $N$ и $M$ ($1 \le N, M \le 100$). Далее в $M$ строках по $N$ чисел в строке следует матрица $A_{ij}$. В последней строке записаны $N$ чисел $b_j$ — координаты необходимого вектора тяги. Все числа $A_{ij}$ и $b_j$ целые и по модулю не превосходят 100.

## Output

Выведите M действительных чисел $X_1, \ldots, X_M$ с точностью в пять знаков после десятичной точки. $X_i$ должно равняться количеству энергии, которое необходимо подать на $i$-й двигатель. В случае, если ответов несколько, выведите любой.

## Examples

| star2.in | star2.out |
|---|---|
| 4 3<br>2 3 -2 1<br>-1 2 1 3<br>4 2 3 -2<br>3 13 -9 13 | 4.00000 2.00000 -1.00000 |