

1、道路

N 个以 1 ... N 标号的城市通过单向的道路相连:。每条道路包含两个参数：道路的长度和需要为该路付的通行费（以金币的数目来表示）

Bob and Alice 过去住在城市 1.在注意到 Alice 在他们过去喜欢玩的纸牌游戏中作弊后，Bob 和她分手了，并且决定搬到城市 N。他希望能够尽可能快的到那，但是他囊中羞涩。我们希望能够帮助 Bob 找到从 1 到 N 最短的路径，前提是他能够付的起通行费。

时间限制：1000

内存限制：65536

输入

第一行包含一个整数 K, $0 \leq K \leq 10000$, 代表 Bob 能够在他路上花费的最大的金币数。
第二行包含整数 N, $2 \leq N \leq 100$, 指城市的数目。第三行包含整数 R, $1 \leq R \leq 10000$, 指路的数目。接下来的 R 行，每行具体指定几个整数 S, D, L 和 T 来说明关于道路的一些情况，这些整数之间通过空格间隔: S is 道路起始城市, $1 \leq S \leq N$ D is 道路终点城市, $1 \leq D \leq N$ L is 道路长度, $1 \leq L \leq 100$ T is 通行费 (以金币数量形式度量), $0 \leq T \leq 100$ 注意不同的道路可能有相同的起点和终点。

输出

输入结果应该只包括一行，即从城市 1 到城市 N 所需要的最小的路径长度（花费不能超过 K 个金币）。如果这样的路径不存在，结果应该输出 -1。

样例输入

```
5
6
7
1 2 2 3
2 4 3 3
3 4 2 4
1 3 4 1
4 6 2 1
3 5 2 0
5 4 3 2
```

样例输出

```
11
```

2、控制公司

有些公司是其他公司的部分拥有者，因为他们获得了其他公司发行的股票的一部分。例如，福特公司拥有马自达公司 12% 的股票。据说，如果至少满足了以下条件之一，公司 A 就可以控制公司 B 了：

I 公司 A = 公司 B。

I 公司 A 拥有大于 50% 的公司 B 的股票。

I 公司 A 控制 $K (K \geq 1)$ 个公司，记为 $C_1, \dots, C_K$ ，每个公司 C_i 拥有 $x_i\%$ 的公司 B 的股票，并且 $x_1 + \dots + x_K > 50\%$ 。（ps：A 可以控制自己，即 C_i 可以为 A）

你将被给予一系列的三对数 (i, j, p) ，表明公司 i 拥有公司 j 的 p% 的股票。计算所有的数对 (h, s) ，表明公司 h 控制公司 s。

写一个程序读入三对数 (i, j, p) ，并且找出所有的数对 (h, s) ，使得公司 h 控制公司 s。

时间限制：1000

内存限制：65536

输入

第一行：N，表明接下来三对数的数量。第二行到第 N+1 行：每行三个整数作为一个三对数 (i, j, p) ，如上文所述。 $1 \leq j \leq 100, N, P \leq 100$

输出

输出零个或更多的控制其他公司的公司。每行包括两个整数表明序号为第一个整数的公司控制了序号为第二个整数的公司。将输出的每行以第一个数字升

序排列（并且第二个数字也升序排列来避免并列）。请不要输出控制自己的公司。

样例输入

```
3
1 2 80
2 3 80
3 1 20
```

样例输出

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

3、发现它，抓住它

一个城市中有两个犯罪团伙 A 和 B，你需要帮助警察判断任意两起案件是否是同一个犯罪团伙所为，警察所获得的信息是有限的。假设现在有 N 起案件（ $N \leq 100000$ ），编号为 1 到 N ，每起案件由团伙 A 或团伙 B 所为。你将按时间顺序获得 M 条信息（ $M \leq 100000$ ），这些信息分为两类：

1. D [a] [b]

其中[a]和[b]表示两起案件的编号，这条信息表明它们属于不同的团伙所为

2. A [a] [b]

其中[a]和[b]表示两起案件的编号，这条信息需要你回答[a]和[b]是否是同一个团伙所为。注意你获得信息的时间是有先后顺序的，在回答的时候只能根据已经接收到的信息做出判断。

时间限制：1000

内存限制：65536

输入

第一行是测试数据的数量 T ($1 \leq T \leq 20$)。 每组测试数据的第一行包括两个数 N 和 M ，分别表示案件的数量和信息数量，其后 M 行表示按时间顺序收到的 M 条信息。

输出

对于每条需要回答的信息，你需要输出一行答案。如果是同一个团伙所为，回答 "In the same gang."，如果不是，回答 "In different gangs."，如果不确定，回答 "Not sure yet."。

样例输入

```
1
5 5
A 1 2
D 1 2
A 1 2
D 2 4
A 1 4
```

样例输出

```
Not sure yet.
In different gangs.
In the same gang.
```

4、青蛙的约会

两只青蛙在网上相识了，它们聊得很开心，于是觉得很有必要见一面。它们很高兴地发现它们住在同一条纬度线上，于是它们约定各自朝西跳，直到碰面为止。可是它们出发之前忘记了一件很重要的事情，既没有问清楚对方的特征，也没有约定见面的具体位置。不过青蛙们都是很乐观的，它们觉得只要一直朝着某个方向跳下去，总能碰到对方的。但是除非这两只青蛙在同一时间跳到同一点上，不然是永远都不可能碰面的。为了帮助这两只乐观的青蛙，你被要求写一个程序来判断这两只青蛙是否能够碰面，会在什么时候碰面。我们把这两只青蛙分别叫做青蛙 A 和青蛙 B，并且规定纬度线上东经 0 度处为原点，由东往西为正方向，单位长度 1 米，这样我们就得到了一条首尾相接的数轴。设青蛙 A 的出发点坐标是 x ，青蛙 B 的出发点坐标是 y 。青蛙 A 一次能跳 m 米，青蛙 B 一次能跳 n 米，两只青蛙跳一次所花费的时间相同。纬度线总长 L 米。现在要你求出它们跳了几次以后才会碰面。

时间限制：1000

内存限制：65536

输入

输入只包括一行 5 个整数 x, y, m, n, L ，其中 $x \neq y < 2000000000$ ， $0 < m, n < 2000000000$ ， $0 < L < 2100000000$ 。

输出

输出碰面所需要的跳跃次数，如果永远不可能碰面则输出一行 "Impossible"

样例输入

```
1 2 3 4 5
```

样例输出

```
4
```