# Skyscraper Floors

What a great idea it is to build skyscrapers! Using not too large area of land, which is very expensive in many cities today, the skyscrapers offer an extremely large utility area for flats or offices. The only disadvantage is that it takes too long to get to the upper floors. Of course these skyscrapers have to be equiped not only with a stairway but also with several elevators. But even using ordinary elevators is very slow. Just imagine you want to get from the very top floor to the base floor and many other people on other floors want the same. As a result the elevator stops on almost every floor and since its capacity is limited and the elevator is already full from the upper floors, most stops are useless and just cause a delay. If there are more elevators in the skyscrapers, this problem is a little bit eliminated but still not completely. Most people just press all the buttons of all the elevators and then take the first one so that all elevators will stop on the floor anyway.

However, the solution exists as we shall see. The Antique Comedians of Midilesia headquarters reside in a skyscraper with a very special elevator system. The elevators do not stop on every floor but only on every $X$-th floor. Moreover each elevator can go just to a certain floor $Y$ (called starting floor) and cannot go any lower. There is one high-capacity elevator which can stop on every elevator's starting floor.

The ACM has a big problem. The headquarters should be moved to another office this week, possibly on a different floor. Unfortunately, the high-capacity elevator is out of order right now so it is not always possible to go to the base floor. One piece of furniture cannot be moved using the stairway because it is too large to pass through the stairway door. You are to write a program that decides whether it is possible to move a piece of furniture from the original office to the other.

## Input

The input consists of $N$ cases. The first line contains only one positive integer $N$. Then follow the cases. Each case starts with a line containing four integers $F$, $E$, $A$, $B$, where $F$, $1 \leq F < 50000000$ determines the number of floors in the skyscraper (this means that there are floors 0 to $F$-1), $E$, $0 < E < 100$ is the number of elevators and $A$, $B$, $0 \leq A, B < F$ are numbers of the two floors between which the piece of furniture should be moved. Then follow $E$ lines. Each of them contains description of one elevator. There are exactly two integers $X$ and $Y$, $X > 0$, $Y \geq 0$ at each line. $Y$ determines, that the elevator starts on the $Y$-th floor and $X$ determines, that it stops on every $X$-th floor, e.g. for $X = 3$, $Y = 7$ the elevator stops on floors 7, 10, 13, 16, etc.).

## Output

For each case, print exactly one line. If floor $B$ is reachable from floor $A$ not using the stairway, print the sentence `` `It is possible to move the furniture.' ``, otherwise print `` `The furniture cannot be moved.' ``.

## Sample Input

```
2
22 4 0 6
3 2
4 7
```

# 742 Domino Game

Let us consider the following version of the Domino game:

- A domino piece has two sides, each of them numbered from 0 to 6.
- The game is played with 28 pieces. A piece having both sides with the same number is called a double piece.
- We will consider a two-player version of the game: each player receives a set of pieces and both sets are equal in size.
- The player who received the largest-valued double piece begins the game placing down this piece, thus starting a configuration with two extremes.
- Each player, in turn, has to place down one of his (her) pieces, always at one of the extremes of the current configuration, if and only if that piece has one of its sides matching the number at the outer side of the extreme piece. The piece is placed so that the sides with matching numbers are adjacent to each other. No special treatment is given to a double piece in that regard: it is placed down with one of its matching sides adjacent to the extreme piece.
- Whenever a player has no piece matching one of the extremes, then he (she) passes his (her) turn.
- The winner is that player who first places down all of his pieces.

Suppose the two players are named Red and Green. Given initial sets of pieces assigned to each of Red and Green, your problem is to determine whether (i) only one of them can win, or (ii) both of them can win, or (iii) none of them can win the game. **Remember that this is not a problem of finding winning strategies. You just have to find out the winning possibility of each player.**

## Input

The input file may contain several instances of the problem. Each instance consists of one line in the following format:

```
n r11 r12 r21 r22 ... rn1 rn2 g11 g12 ... gn1 gn2
```

where:

- `n` is the size of the subsets assigned to Red and Green; you may assume that $1 \leq n \leq 14$;
- `ri1 ri2` is the $i$-th piece assigned to player Red;
- `gi1 gi2` is the $i$-th piece assigned to player Green;
- no pieces are repeated;
- at least one player is assigned a double piece;
- each of `ri1 ri2 gi1 gi2` is an integer number from 0 to 6.

An arbitrary number of blank spaces may separate a number from its neighbours in the input line of an instance. Consecutive instances are NOT separated by blank lines. The last line has $n = 0$.

# Problem E
# "Highways"

The island nation of Flatopia is perfectly flat. Unfortunately, Flatopia has a very poor system of public highways. The Flatopian government is aware of this problem and has already constructed a number of highways connecting some of the most important towns. However, there are still some towns that you can't reach via a highway. It is necessary to build more highways so that it will be possible to drive between any pair of towns without leaving the highway system.

Flatopian towns are numbered from 1 to $N$ and town $i$ has a position given by the Cartesian coordinates $(x_i, y_i)$. Each highway connects exaclty two towns. All highways (both the original ones and the ones that are to be built) follow straight lines, and thus their length is equal to Cartesian distance between towns. All highways can be used in both directions. Highways can freely cross each other, but a driver can only switch between highways at a town that is located at the end of both highways.

The Flatopian government wants to minimize the cost of building new highways. However, they want to guarantee that every town is highway-reachable from every other town. Since Flatopia is so flat, the cost of a highway is always proportional to its length. Thus, the least expensive highway system will be the one that minimizes the total highways length.

## Input

The first line of the input consists of an integer indicating the number of test cases in the input. Then there's a blank line and the datasets separated by a blank line.

Each test case consists of two parts. The first part describes all towns in the country, and the second part describes all of the highways that have already been built.

The first line of the test case contains a single integer $N$ ($1 \le N \le 750$), representing the number of towns. The next $N$ lines each contain two integers, $x_i$ and $y_i$ separated by a space. These values give the coordinates of $i^{\text{th}}$ town (for $i$ from 1 to $N$). Coordinates will have an absolute value no greater than 10000. Every town has a unique location.

The next line contains a single integer $M$ ($0 \le \text{M} \le 1000$), representing the number of existing highways. The next $M$ lines each contain a pair of integers separated by a space. These two integers give a pair of town numbers which are already connected by a highway. Each pair of towns is connected by at most one highway.

## Output

Write to the output file a single line for each new highway that should be built in order to connect all towns with minimal possible total length of new highways. Each highway should be presented by printing town numbers that this highway connects, separated by a space.

If no new highways need to be built (all towns are already connected), then the output file should contain a line with the sentence "`No new highways need`".

Print a blank line between test cases.

## Sample input

```
1

9
1 5
0 0
3 2
4 5
5 1
0 4
5 2
1 2
5 3
3
1 3
9 7
1 2
```

## Sample output for the sample input

```
1 6
3 7
4 9
5 7
8 3
```

# Problem G - Monitoring the Amazon

**Time Limit: 5 seconds**

**Memory Limit: 1 Mb**

# Background

A network of autonomous, battery-powered, data acquisition stations has been installed to monitor the climate in the region of Amazon. An order-dispatch station can initiate transmission of instructions to the control stations so that they change their current parameters. To avoid overloading the battery, each station (including the order-dispatch station) can only transmit to two other stations. The destinataries of a station are the two closest stations. In case of draw, the first criterion is to chose the westernmost (leftmost on the map), and the second criterion is to chose the southernmost (lowest on the map).

# The Problem

You are commissioned by Amazon State Government to write a program that decides if, given the localization of each station, messages can reach all stations.

# Input

The input consists of an integer $N$, followed by $N$ pairs of integers $Xi$, $Yi$, indicating the localization coordinates of each station. The first pair of coordinates determines the position of the order-dispatch station, while the remaining $N-1$ pairs are the coordinates of the other stations. The following constraints are imposed: $-20 \leq Xi, Yi \leq 20$, and $1 \leq N \leq 1000$. The input is terminated with $N = 0$.

# Output

For each given expression, the output will echo a line with the indicating if all stations can be reached or not (see sample output for the exact format).