653 Gizilch

The game of gizilch has very simple rules. First 100 grapes are labeled, in nontoxic ink, with the numbers 1 to 100. Then, with a cry of "GIZILCH!", the referee fires the grapes up into the air with a giant gizilcher. The two players, who each start with a score of "1", race to eat the falling (or, shortly thereafter, fallen) grapes and, at the same time, multiply their scores by the numbers written on the grapes they eat. After a minute, the hungry squirrels are let loose to finish the remaining grapes, and each contestant reports his score, the product of the numbers on the grapes he's eaten. The unofficial winner is the player who announces the highest score.

Inevitably, though, disputes arise, and so the official winner is not determined until the disputes are resolved. The player who claims the lower score is entitled to challenge his opponent's score. The player with the lower score is presumed to have told the truth, because if he were to lie about his score, he would surely come up with a bigger better lie. The challenge is upheld if the player with the higher score has a score that cannot be achieved with grapes not eaten by the challenging player. So, if the challenge is successful, the player claiming the lower score wins.

So, for example, if one player claims 343 points and the other claims 49, then clearly the first player is lying; the only way to score 343 is by eating grapes labeled 7 and 49, and the only way to score 49 is by eating a grape labeled 49. Since each of two scores requires eating the grape labeled 49, the one claiming 343 points is presumed to be lying.

On the other hand, if one player claims 162 points and the other claims 81, it is possible for both to be telling the truth (e.g. one eats grapes 2, 3 and 27, while the other eats grape 81), so the challenge would not be upheld.

Unfortunately, anyone who is willing to referee a game of gizilch is likely to have himself consumed so many grapes (in a liquid form) that he or she could not reasonably be expected to perform the intricate calculations that refereeing requires. Hence the need for you, sober programmer, to provide a software solution.

Input

Pairs of unequal, positive numbers, with each pair on a single line, that are claimed scores from a game of gizilch.

Output

Numbers, one to a line, that are the winning scores, assuming that the player with the lower score always challenges the outcome.

Sample Input

```
343 49
3599 610
62 36
```

Sample Output

```
49
610
62
```

681 Convex Hull Finding

Given a single connected contour, which is either convex or non-convex (concave), use any algorithm to find its **Convex Hull**, i.e., the smallest convex contour enclosing the given shape. If the given contour is convex, then its convex hull is the original contour itself. The maximal size of the shape is 512×512, and the maximal number of the vertices of the shape is 512. Write a program to read the input data (the given shapes) from a disk file, implement your convex hull finding algorithm, and then output the shape data of the results to the standard output.

Input

The order of the vertices is counterclockwise in $X - Y$ Cartesian Plane (if you consider the origin of the display window is on the upper-left corner, then the orientation of the vertices is clockwise), and none of the neighboring vertices are co-linear. Since all the shapes are closed contours, therefore, the last vertex should be identical to the first vertex. There are several sets of data within a given data file. The negative number `-1` is used to separate the data set.

Line Number	**Data in the File**	**Explanation**
1	K	a positive integer showing how many sets of data in this file
2	N	a positive integer showing the number of vertices for the shape
3	X_1 Y_1	two positive integers for the first vertex (X_1, Y_1)
4	X_2 Y_2	two positive integers for the next neighboring vertex (X_2, Y_2)
...		
$N+2$	X_N Y_N	two positive integers for the last vertex (X_N, Y_N)
$N+3$	`-1`	Delimiter
$N+4$	M	a positive integer showing the number of vertices for the next shape
$N+5$	XX_1 YY_1	two positive integers for the first vertex
...		

Note: Please note that the **Line Number**, **Data in the File** and **Explanation** are not given in the file. They are shown here only to assist you in reading the data.

Output

Output the convex hull of all K input shapes to the standard output. The data format should be the same as the input file. In addition, the vertex with the smallest Y value should be the first point and if there are points with the same Y value, then the smallest X value within those points should be the first point.

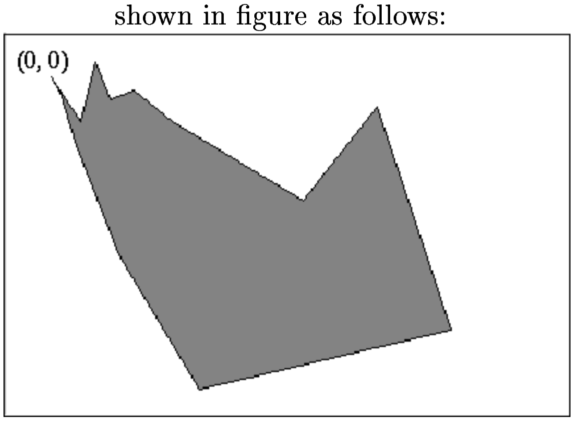
Sample Input

```
3
15
30 30
50 60
60 20
70 45
86 39
112 60
200 113
250 50
300 200
130 240
76 150
47 76
36 40
33 35
30 30
-1
12
50 60
60 20
70 45
100 70
125 90
200 113
250 140
180 170
105 140
79 140
60 85
50 60
-1
6
60 20
250 140
180 170
79 140
50 60
60 20
```

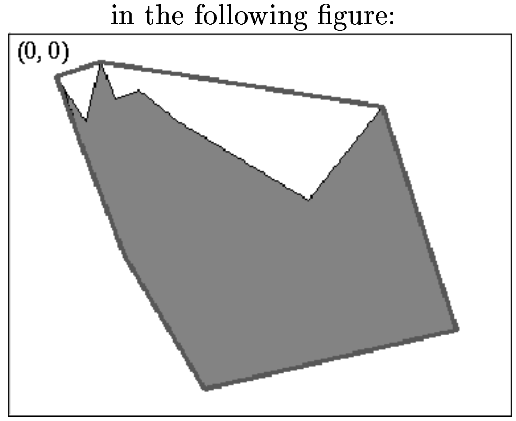
Sample Output

```
3
8
60 20
250 50
300 200
130 240
76 150
47 76
30 30
60 20
-1
6
60 20
250 140
180 170
79 140
50 60
60 20
-1
6
60 20
250 140
180 170
79 140
50 60
60 20
```

The contour shape of the first data set is shown in figure as follows:



The convex hull of the above shape is shown in the following figure:



702 The Vindictive Coach

The coach of a football team, after suffering for years the adverse comments of the media about his tactics, decides to take his revenge by presenting his players in a line-up in such a way that the TV cameras would be compelled to zigzag in a ridiculous bobbing motion, by alternating taller and shorter players. However, the team captain objects that he must be the first of the line by protocolary reasons,and that he wants to be seen in the best possible light: that is, he should not have a taller colleague nest to him unless there is no alternative (everyone else is taller than him). Even in this case, the height difference should be as small as possible, while maintaining the zigzag arrangement of the line.

With this condition the coach addresses an expert in computation (i.e. you) to help him find the number of different alignments he may make, knowing that all players have a different height. They are always numbered by stature starting by 1 as the shortest one. Of course the number of players may be arbitrary, provided it does not exceed 22.

Input

It is a set of lines, each of which contains two positive integers N and m separated by a blank space. $N(\leq 22)$ represents the number of players in the line-up and m the captain's number, who as told is always the first of the line.

Output

For every line of the input a line with positive integer indicating the number of possible alignments under the above conditions.

Sample Input

```
3 1
3 3
4 1
```

Sample Output

```
1
1
1
```

718 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
13 6
10 0
1000 2 500 777
2 0
2 1
```

Sample Output

```
It is possible to move the furniture.
The furniture cannot be moved.
```