A. Balls Rearrangement

Bob has N balls and A boxes. He numbers the balls from 0 to N-1, and numbers the boxes from 0 to A-1. To find the balls easily, he puts the ball numbered x into the box numbered a if x = a mod A. Some day Bob buys B new boxes, and he wants to rearrange the balls from the old boxes to the new boxes. The new boxes are numbered from 0 to B-1. After the rearrangement, the ball numbered x should be in the box number b if x = b mod B.

This work may be very boring, so he wants to know the cost before the rearrangement. If he moves a ball from the old box numbered a to the new box numbered b, the cost he considered would be |a-b|. The total cost is the sum of the cost to move every ball, and it is what Bob is interested in now.

Input

The first line of the input is an integer T, the number of test cases.(0<T<=50)

Then T test case followed. The only line of each test case are three integers N, A and B. (1<=N<=1000000000, 1<=A,B<=100000).

Output

For each test case, output the total cost.

Sample Input

```
3
1000000000 1 1
8 2 4
11 5 3
```

Sample Output

```
0
8
16
```

B. Warm up

N planets are connected by M bidirectional channels that allow instant transportation. It's always possible to travel between any two planets through these channels.

If we can isolate some planets from others by breaking only one channel , the channel is called a bridge of the transportation system.

People don't like to be isolated. So they ask what's the minimal number of bridges they can have if they decide to build a new channel.

Note that there could be more than one channel between two planets.

Input

The input contains multiple cases.

Each case starts with two positive integers N and M , indicating the number of planets and the number of channels.

(2<=N<=200000, 1<=M<=1000000)

Next M lines each contains two positive integers A and B, indicating a channel between planet A and B in the system. Planets are numbered by 1..N.

A line with two integers '0' terminates the input.

Output

For each case, output the minimal number of bridges after building a new channel in a line.

Sample Input

```
4 4
1 2
1 3
1 4
2 3
0 0
```

Sample Output

```
0
```

C. Vases and Flowers

Alice is so popular that she can receive many flowers everyday. She has N vases numbered from 0 to N-1. When she receive some flowers, she will try to put them in the vases, one flower in one vase. She randomly choose the vase A and try to put a flower in the vase. If the there is no flower in the vase, she will put a flower in it, otherwise she skip this vase. And then she will try put in the vase A+1, A+2, ..., N-1, until there is no flower left or she has tried the vase N-1. The left flowers will be discarded. Of course, sometimes she will clean the vases. Because there are too many vases, she randomly choose to clean the vases numbered from A to B(A <= B). The flowers in the cleaned vases will be discarded.

Input

The first line contains an integer T, indicating the number of test cases.

For each test case, the first line contains two integers N(1 < N < 50001) and M(1 < M < 50001). N is the number of vases, and M is the operations of Alice. Each of the next M lines contains three integers. The first integer of one line is K(1 or 2). If K is 1, then two integers A and F follow. It means Alice receive F flowers and try to put a flower in the vase A first. If K is 2, then two integers A and B follow. It means the owner would like to clean the vases numbered from A to B(A <= B).

Output

For each operation of which K is 1, output the position of the vase in which Alice put the first flower and last one, separated by a blank. If she can not put any one, then output 'Can not put any one.'. For each operation of which K is 2, output the number of discarded flowers.

Output one blank line after each test case.

Sample Input

```
2
10 5
1 3 5
2 4 5
1 1 8
2 3 6
1 8 8
10 6
1 2 5
2 3 4
1 0 8
2 2 5
1 4 4
1 2 3
```

Sample Output

```
[pre]3 7
2
1 9
4
Can not put any one.

2 6
2
0 9
4
4 5
2 3

[/pre]
```

D. Vases and Flowers

Alice is so popular that she can receive many flowers everyday. She has N vases numbered from 0 to N-1. When she receive some flowers, she will try to put them in the vases, one flower in one vase. She randomly choose the vase A and try to put a flower in the vase. If the there is no flower in the vase, she will put a flower in it, otherwise she skip this vase. And then she will try put in the vase A+1, A+2, ..., N-1, until there is no flower left or she has tried the vase N-1. The left flowers will be discarded. Of course, sometimes she will clean the vases. Because there are too many vases, she randomly choose to clean the vases numbered from A to B(A <= B). The flowers in the cleaned vases will be discarded.

Input

The first line contains an integer T, indicating the number of test cases.

For each test case, the first line contains two integers N(1 < N < 50001) and M(1 < M < 50001). N is the number of vases, and M is the operations of Alice. Each of the next M lines contains three integers. The first integer of one line is K(1 or 2). If K is 1, then two integers A and F follow. It means Alice receive F flowers and try to put a flower in the vase A first. If K is 2, then two integers A and B follow. It means the owner would like to clean the vases numbered from A to B(A <= B).

Output

For each operation of which K is 1, output the position of the vase in which Alice put the first flower and last one, separated by a blank. If she can not put any one, then output 'Can not put any one.'. For each operation of which K is 2, output the number of discarded flowers.

Output one blank line after each test case.

Sample Input

```
2
10 5
1 3 5
2 4 5
1 1 8
2 3 6
1 8 8
10 6
1 2 5
2 3 4
1 0 8
2 2 5
1 4 4
1 2 3
```

Sample Output

```
[pre]3 7
2
1 9
4
Can not put any one.

2 6
2
0 9
4
4 5
2 3

[/pre]
```

E. Play with Sequence

ivankevin loves sequences.

A sequence of n elements is called a **SWITCH SEQUENCE** iff:

Each element is a **2-switch**, where a 2-switch is the replacement of a pair of edges **xy** & **zw** in a simple graph by the edges **yz** and **wx**, given that **yz** and **wx** did not appear in the graph originally.

Now give two simple graphs G and H with same vertex set V, ivankevin wants to know whether there is a switch sequence transforms G into H.

Input

There are multiple test cases.

Each case contains 1 + 2n lines. The positive integer n(0<n<101) is given in the first line. The next 2n lines are the matrix which describe the two graphs G and H. The j elements in the i rows is either 0 or 1. If the node i and the node j has a edge in the graph. The elements will be 1. Otherwise is 0. It’s guaranteed that the matrix is symmetry and the i elements in the i row is always 0. See sample input for more details.

Output

For each case, first output an integer L in a line, which should represent the length of the switch sequence you found.

If no switch sequence can transform G into H, then you should print L as -1 and no more lines. Otherwise output L lines follow. Each of the L lines should contains four distinct integers x, y, z and w, indicating a 2-switch on G, as described.

For some reason, if L is equal to or larger than 1000000, ivankevin would think the switch sequence you found is so ugly that it is ungelieveble and unacceptable. Otherwise, if L < 1000000 and the sequence finally transform G into H, he will accept your answer.

Sample Input

```
4
0 1 0 0
1 0 0 0
0 0 0 1
0 0 1 0
0 0 1 0
0 0 0 1
1 0 0 0
0 1 0 0
4
0 1 0 0
1 0 0 0
0 0 0 1
0 0 1 0
0 1 1 0
1 0 0 0
1 0 0 0
0 0 0 0
```

Sample Output

```
1
0 1 3 2
-1
```

F. Game

Nowadays, there are more and more challenge game on TV such as 'Girls, Rush Ahead'. Now, you participate int a game like this. There are N rooms. The connection of rooms is like a tree. In other words, you can go to any other room by one and only one way. There is a gift prepared for you in Every room, and if you go the room, you can get this gift. However, there is also a trap in some rooms. After you get the gift, you may be trapped. After you go out a room, you can not go back to it any more. You can choose to start at any room ,and when you have no room to go or have been trapped for C times, game overs. Now you would like to know what is the maximum total value of gifts you can get.

Input

The first line contains an integer T, indicating the number of testcases.

For each testcase, the first line contains one integer N(2 <= N <= 50000), the number rooms, and another integer C(1 <= C <= 3), the number of chances to be trapped. Each of the next N lines contains two integers, which are the value of gift in the room and whether have trap in this rooom. Rooms are numbered from 0 to N-1. Each of the next N-1 lines contains two integer A and B(0 <= A,B <= N-1), representing that room A and room B is connected.

All gifts' value are bigger than 0.

Output

For each testcase, output the maximum total value of gifts you can get.

Sample Input

```
2
3 1
23 0
12 0
123 1
0 2
2 1
3 2
23 0
12 0
123 1
0 2
2 1
```

Sample Output

```
146
158
```

G. Weapon

Doctor D. are researching for a horrific weapon. The muzzle of the weapon is a circle. When it fires, rays form a cylinder that runs through the circle verticality in both side. If one cylinder of rays touch another, there will be an horrific explosion. Originally, all circles can rotate easily. But for some unknown reasons they can not rotate any more. If these weapon can also make an explosion, then Doctor D. is lucky that he can also test the power of the weapon. If not, he would try to make an explosion by other means. One way is to find a medium to connect two cylinder. But he need to know the minimum length of medium he will prepare. When the medium connect the surface of the two cylinder, it may make an explosion.

Input

The first line contains an integer T, indicating the number of testcases. For each testcase, the first line contains one integer N(1 < N < 30), the number of weapons. Each of the next 3N lines contains three float numbers. Every 3 lines represent one weapon. The first line represents the coordinates of center of the circle, and the second line and the third line represent two points in the circle which surrounds the center. It is supposed that these three points are not in one straight line. All float numbers are between -1000000 to 1000000.

Output

For each testcase, if there are two cylinder can touch each other, then output 'Lucky', otherwise output then minimum distance of any two cylinders, rounded to two decimals, where distance of two cylinders is the minimum distance of any two point in the surface of two cylinders.

Sample Input

```
3
3
0 0 0
1 0 0
0 0 1
5 2 2
5 3 2
5 2 3
10 22 -2
11 22 -1
11 22 -3
3
0 0 0
1 0 1.5
1 0 -1.5
112 115 109
114 112 110
109 114 111
-110 -121 -130
-115 -129 -140
-104 -114 -119.801961
3
0 0 0
1 0 1.5
1 0 -1.5
112 115 109
114 112 110
109 114 111
-110 -121 -130
-120 -137 -150
-98 -107 -109.603922
```

Sample Output

```
Lucky
2. 32
Lucky
```

H. Palindrome Sub-Array

A palindrome sequence is a sequence which is as same as its reversed order. For example, 1 2 3 2 1 is a palindrome sequence, but 1 2 3 2 2 is not. Given a 2-D array of N rows and M columns, your task is to find a maximum sub-array of P rows and P columns, of which each row and each column is a palindrome sequence.

Input

The first line of input contains only one integer, T, the number of test cases. Following T blocks, each block describe one test case.

There is two integers N, M (1<=N, M<=300) separated by one white space in the first line of each block, representing the size of the 2-D array.

Then N lines follow, each line contains M integers separated by white spaces, representing the elements of the 2-D array. All the elements in the 2-D array will be larger than 0 and no more than 31415926.

Output

For each test case, output P only, the size of the maximum sub-array that you need to find.

Sample Input

```
1
5 10
1 2 3 3 2 4 5 6 7 8
1 2 3 3 2 4 5 6 7 8
1 2 3 3 2 4 5 6 7 8
1 2 3 3 2 4 5 6 7 8
1 2 3 9 10 4 5 6 7 8
```

Sample Output

```
4
```

I. Warm up 2

Some 1×2 dominoes are placed on a plane. Each dominoe is placed either horizontally or vertically. It's guaranteed the dominoes in the same direction are not overlapped, but horizontal and vertical dominoes may overlap with each other. You task is to remove some dominoes, so that the remaining dominoes do not overlap with each other. Now, tell me the maximum number of dominoes left on the board.

Input

There are multiple input cases.

The first line of each case are 2 integers: n(1 <= n <= 1000), m(1 <= m <= 1000), indicating the number of horizontal and vertical dominoes.
Then n lines follow, each line contains 2 integers x (0 <= x <= 100) and y (0 <= y <= 100), indicating the position of a horizontal dominoe. The dominoe occupies the grids of (x, y) and (x + 1, y).

Then m lines follow, each line contains 2 integers x (0 <= x <= 100) and y (0 <= y <= 100), indicating the position of a horizontal dominoe. The dominoe occupies the grids of (x, y) and (x, y + 1).

Input ends with n = 0 and m = 0.

Output

For each test case, output the maximum number of remaining dominoes in a line.

Sample Input

```
2 3
0 0
0 3
0 1
1 1
1 3
4 5
0 1
0 2
3 1
2 2
0 0
1 0
2 0
4 1
3 2
0 0
```

Sample Output

```
4
6
```

J. Fruit Ninja Extreme

Cut or not to cut, it is a question.

In Fruit Ninja, comprising three or more fruit in one cut gains extra bonuses. This kind of cuts are called bonus cuts.

Also, performing the bonus cuts in a short time are considered continual, iff. when all the bonus cuts are sorted, the time difference between every adjacent cuts is no more than a given period length of W.

As a fruit master, you have predicted the times of potential bonus cuts though the whole game. Now, your task is to determine how to cut the fruits in order to gain the most bonuses, namely, the largest number of continual bonus cuts.

Obviously, each fruit is allowed to cut at most once. i.e. After previous cut, a fruit will be regarded as invisible and won't be cut any more.

In addition, you must cut all the fruit altogether in one potential cut. i.e. If your potential cut contains 6 fruits, 2 of which have been cut previously, the 4 left fruits have to be cut altogether.

Input

There are multiple test cases.

The first line contains an integer, the number of test cases.

In each test case, there are three integer in the first line: N(N<=30), the number of predicted cuts, M(M<=200), the number of fruits, W(W<=100), the time window.

N lines follows.

In each line, the first integer Ci(Ci<=10) indicates the number of fruits in the i-th cuts.

The second integer Ti(Ti<=2000) indicate the time of this cut. It is guaranteed that every time is unique among all the cuts.

Then follow Ci numbers, ranging from 0 to M-1, representing the identifier of each fruit. If two identifiers in different cuts are the same, it means they represent the same fruit.

Output

For each test case, the first line contains one integer A, the largest number of continual bonus cuts.

In the second line, there are A integers, K1, K2, ..., K_A, ranging from 1 to N, indicating the (Ki)-th cuts are included in the answer. The integers are in ascending order and each separated by one space. If there are multiple best solutions, any one is accepted.

Sample Input

```
1
4 10 4
3 1 1 2 3
4 3 3 4 6 5
3 7 7 8 9
3 5 9 5 4
```

Sample Output

```
3
1 2 3
```