Greetings From Globussoft

- Given below are 5 Programming questions, you have to solve any 3 out of 5 questions.

- These 5 questions you can attempt in any technology like C/C++, java, .Net, PHP

- To solve these 3 questions you've max. 3 hours.

- While Solving these questions you are not allowed to use any **Search Engine** like Google, Yahoo, Bing …

All the best for your test

Globussoft

QUESTION – 1

There are given *n* men and *n* women. Each woman ranks all men in order of her preference (her first choice, her second choice, and so on). Similarly, each man sorts all women according to his preference. The goal is to arrange *n* marriages in such a way that if a man *m* prefers some woman *w* more than his wife, then *w* likes her husband more than *m*. In this way, no one leaves his partner to marry somebody else. This problem always has a solution and your task is to find one.

Input

The first line contains a positive integer *t*<=100 indicating the number of test cases. Each test case is an instance of the stable marriage problem defined above. The first line of each test case is a positive integer *n*<=500 (the number of marriages to find). The next *n* lines are the woman's preferences: *i*th line contains the number *i* (which means that this is the list given by the *i*th woman) and the numbers of men (the first choice of *i*th woman, the second choice,...). Then, the men's preferences follow in the same format.

Output

For each test case print *n* lines, where each line contains two numbers *m* and *w*, which means that the man number *m* and the woman number *w* should get married.

Example

Input:

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

Output:

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

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

QUESTION – 2

A *partition* of positive integer m into n components is any sequence $a_1,...,a_n$ of positive integers such that $a_1+...+a_n=m$ and $a_1<=a_2<=...<=a_n$. Your task is to determine the partition, which occupies the k-th position in the lexicographic order of all partitions of m into n components.

The lexicographic order is defined as follows: sequence $a_1,...,a_n$ comes before $b_1,...,b_n$ iff there exists such an integer i,$1<=i<=n$, that $a_j=b_j$ for all j, $1<= j< i$, and $a_i< b_i$.

Input

The input begins with the integer t, the number of test cases. Then t test cases follow.

For each test case the input consists of three lines, containing the positive integers m, n and k respectively ($1<=n<= 10$, $1<= m<=220$, k is not larger than the number of partitions of m into n components).

Output

For each test case output the ordered elements of the sought partition, separated by spaces.

Example

Sample input:

```
1
9
4
3
```

Sample output:

```
1 1 3 4
```

QUESTION – 3

Little Johnny decided he needed to stick an open metal box to the floor in the hall of his parents' house, so that all guests coming in would trip on it. He knew that as soon as his parents saw what he had done, they would try to remove it, and he wasn't going to stand for this. So, he chose the strongest glue in his possession and left lots of dabs of it on the floor (from our point of view, these can be regarded as points). Now, the only question that remained was how to stick the box onto the floor. Johnny is very particular about the way he does this: the box is always stuck face down, so that it only touches the floor on the four edges of the rectangle that forms its base. He would like each of these edges to make contact with at least two dabs of glue. Furthermore, he

doesn't want any of the dabs to stay outside the box, since this would ruin the fun (there is no way you can trip someone up, if you've glued them to the floor, is there?).

Obviously, Johnny can sometimes reach his objective in more than one way (especially since he has prepared boxes of all possible dimensions for his act of mischief). Depending on how he does this, a different section of floor will be covered by the box. Determine in how many ways Johnny can choose the section of floor to be covered by the box when gluing.

Input

The input begins with the integer t, the number of test cases. Then t test cases follow.

The first line of each test case contains positive integer n<=10000 - the number of dabs of glue on the floor. The next n lines contain two integers, x y (-15000<=x,y<=15000), representing the x and y coordinates of the dabs (given in the order in which they were placed by Johnny ;).

Output

For each test case output the number of different sections of floor Johnny may choose to cover (possibly 0).

Example

Sample input:

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

Sample output:

```
2
```

QUESTION – 4

Bob has a difficult job. He must distribute advertising booklets for extra school activities in different schools. The booklets have different number of pages. Bob has a list with the number of pages of each booklet and the number of schools that he must visit. He has to distribute the booklets such that each school gets a number of booklets equal to either the lower integer part (LIP),or the upper integer part (UIP) of the number of booklets divided by the number of schools. Poor Bob must obey other rules too. He must distribute all the **UIP** number of booklets first and then the **LIP** number of booklets.

Any booklet **A** that is distributed to a school **S_i** must have fewer or at most an equal number of pages that any other booklet B that is distributed to a school **S_j**, if **S_i** gets the booklets before **S_j** (i.e if **i** < **j** then **pages(A)** <= **pages(B)**). When Bob distributes the booklets to a school he must distribute them in the same relative order in which they are on his list.

Moreover, he must distribute them very fast. When he comes back to the advertising company his boss verifies if he accomplished well his task, by asking him the number of pages of the first booklet distributed to a specific school, following the order in which Bob visited the schools (starting with 0). Difficult job, isn't it? Can you help him?

Input

The input starts with a line containing a single integer **t** <= 20, the number of test cases. **t** test cases follow.

Each data set in the input stands for a particular set of bookets. For each set of booklets the input contains the number of schools, the school specified by Bob's boss, the number of booklets (**less than 3000**), the number of pages of each booklet (fits in integer). White spaces can occur freely between the numbers in the input. The input data are correct.

Output

For each set of data the program prints the result to the standard output on a separate line. The solution is represented by the number of pages of the first booklet distributed to the specified school.

Example

Input:

```
1
3
2
7
3 5 9 1 11 14 2
```

Output:

```
11
```

QUESTION – 5

Students of computer science in Bratislava enjoy hiking and camping during their long summer breaks. They love walking silently in the groves, visiting sparkling waterfalls, exploring dark caves, climbing steep hills, or just sleeping in a tent. Some of them already visited all the national parks in Slovakia and nearby countries.

With no more new national parks to visit, frustrated students decided to set up a new national park (NP) by themselves. After long arguing, they finally agreed on the boundary of the NP. Now they want to purchase all the land needed for NP from present owners. Their funds are limited (after all, they are only students), therefore they do not want to buy any land outside the NP.

The NP can be described as a polygon with N vertices. There is a set P of M rectangular plots of land available for sale by their owners. The rectangles are mutually disjoint and axis-parallel. Your task is to decide whether it is possible to purchase subset of plots P exactly covering the proposed NP.

Input

Input file consists of several test cases separated by a blank line. Each test case starts with two integers N and M. Next N lines contain the coordinates of the vertices of the NP. Each of the following M lines describes one plot. For each plot, the coordinates of two opposite corners of the rectangle are given. The values N=0, M=0 end the input and should not be processed. [N, M <= 3000]

Output

For each test case output either 'YES' or 'NO' depending on whether it is possible to set up the NP using P or not.

Example

Input:

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

3 1
0 0
2 2
2 0
0 0 1 1

0 0
```

Output:

```
YES
NO
```