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

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 – 2

A parliamentary election was being held in Byteland. Its enterprising and orderly citizens decided to limit the entire election campaign to a single dedicated wall, so as not to ruin the panorama with countless posters and billboards. Every politician was allowed to hang exactly one poster on the wall. All posters extend from top to bottom, but are hung at different points of the wall, and may be of different width. The wall is divided horizontally into sections, and a poster completely occupies two or more adjacent sections.

With time, some of the posters were covered (partially or completely) by those of other politicians. Knowing the location of all the posters and the order in which they were hung, determine how many posters have at least one visible section in the end.

Input

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

Each test case begins with a line containing integer n - the number of posters ($1<=n<=40000$). Then n lines follow, the i-th ($1<=i<=n$) containing exactly two integers l_i r_i, denoting the numbers of the leftmost and rightmost sections covered by the i-th poster ($1<=l_i< r_i<= 10^7$). The input order corresponds to the order of hanging posters.

Output

For each test case output a line containing one integer - the number of posters with visible sections.

Example

Sample input:

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

Sample output:

```
4
```

QUESTION – 3

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 – 4

You are given two short sequences of numbers, X and Y. Try to determine the minimum number of steps of transformation required to convert sequence X into sequence Y, or determine that such a conversion is impossible.

In every step of transformation of a sequence, you are allowed to replace exactly one occerunce of one of its elements by a sequence of 2 or 3 numbers inserted in its place, according to a rule specified in the input file.

Input

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

For each test case, the first line of input contains four integers - N, M, U, V (1<=N,M<=50). The next two lines of input contain sequences X and Y, consisting of N and M integers respectively. The next U lines contain three integers: *a b c* each, signifying that integer *a* can be converted to the sequence *b c* in one step of transformation. The next V-U lines contain four integers: *a b c d* each, signifying that integer *a* can be converted to the sequence *b c d* in one step of transformation. With the exception of N and M, all integers provided at input are positive and do not exceed 30.

The format of one set of input data is illustrated below.

$N\ M\ U\ V$
$x_1\ x_2\ \ldots\ x_N$
$y_1\ y_2\ \ldots\ y_M$
$a_1\ b_1\ c_1$
$\vdots$
$a_U\ b_U\ c_U$
$a_{U+1}\ b_{U+1}\ c_{U+1}\ d_{U+1}$
$\vdots$
$a_V\ b_V\ c_V\ d_V$

Output

For each test case output -1 if it is impossible to convert sequence X into sequence Y, or the minimum number of steps required to achieve this conversion otherwise.

Example

Sample input:
```
1
3 10 2 3
2 3 1
2 1 1 2 2 1 2 1 2 1
3 1 2
3 3 3
3 1 3 2
```

Sample output:
```
6
```

QUESTION – 5

There are infinitely many coin denominations in the Byteland. They have values of 2^i for i=0,1,2,... . We will say that set of coins c1,c2,...,ck is perfect when it is possible to pay every amount of money between 0 and c1+...+ck using some of them (so {4,2,2,1} is perfect while {8,1} is not). The question is - is it always possible to change given sum n into a perfect set of coins? Of course it is possible ;). Your task will be more complicated: for a sum n you should find minimal number of coins in its perfect representation.

Input

First line of input contains one integer c<=50 - number of test cases. Then c lines follow, each of them consisting of exactly one integer n<=10^1000.

Output

For each test case output minimal number of coins.

Example

```
Input:
5
507
29
8574
233
149

Output:
14
7
21
11
10
```