

Stop the Castle 2

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 1024 megabytes

There are n castles and m obstacles on a chessboard with 10^9 rows and 10^9 columns. Each castle or obstacle occupies exactly one cell and all occupied cells are distinct. Two castles can attack each other, if they're on the same row or the same column, and there are no obstacles or other castles between them. More formally, let (i, j) be the cell on the i -th row and the j -th column. Two castles positioned at (i_1, j_1) and (i_2, j_2) can attack each other, if one of the following conditions is true:

- $i_1 = i_2$ and for all $\min(j_1, j_2) < j < \max(j_1, j_2)$, there is no obstacle or castle positioned at (i_1, j) .
- $j_1 = j_2$ and for all $\min(i_1, i_2) < i < \max(i_1, i_2)$, there is no obstacle or castle positioned at (i, j_1) .

You have to remove k obstacles from the chessboard, but you don't want too many castles to attack each other. Minimize the number of pairs of castles which can attack each other after removing exactly k obstacles from the chessboard.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains three integers n , m , and k ($1 \leq n, m \leq 10^5$, $1 \leq k \leq m$) indicating the number of castles, the number of obstacles, and the number of obstacles you have to remove.

For the following n lines, the i -th line contains two integers r_i and c_i ($1 \leq r_i, c_i \leq 10^9$) indicating that the i -th castle is located on the r_i -th row and the c_i -th column.

For the following m lines, the i -th line contains two integers r'_i and c'_i ($1 \leq r'_i, c'_i \leq 10^9$) indicating that the i -th obstacle is located on the r'_i -th row and the c'_i -th column.

It's guaranteed that the occupied cells are distinct. It's also guaranteed that neither the sum of n nor the sum of m of all test cases will exceed 10^5 .

Output

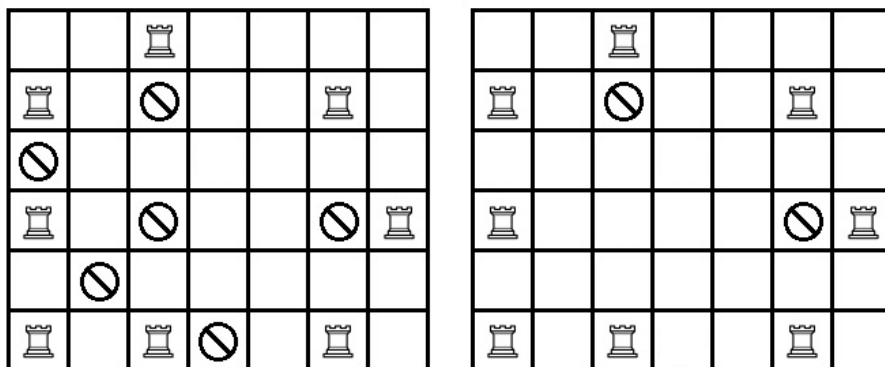
For each test case, first output one line containing one integer indicating the smallest number of pairs of castles which can attack each other after removing exactly k obstacles. Then output another line containing k distinct integers b_1, b_2, \dots, b_k ($1 \leq b_i \leq m$) separated by a space, indicating the indices of obstacles you want to remove. If there are multiple valid answers, you can output any of them.

Example

standard input	standard output
3 8 6 4 1 3 2 1 2 6 4 1 4 7 6 1 6 3 6 6 2 3 3 1 4 3 4 6 5 2 6 4 3 2 1 10 12 10 10 10 11 1 4 1 5 1 3 2 1 1 2 1 2 2 2 3	4 6 3 2 5 2 1 0 1 2

Note

For the first sample test case, the image on the left shows the original chessboard, and the image on the right shows the chessboard after removing 4 obstacles. After removing the obstacles, the pairs of castles which can attack each other are: the 2-nd and the 4-th castles, the 4-th and the 6-th castles, the 6-th and the 7-th castles, the 7-th and the 8-th castles.



For the third sample test case, as there is only 1 castle, there is no pair of castles which can attack each other.