Queens, Knights and Pawns

Input: `qkp.in`
Output: `standard output`

You all are familiar with the famous 8-queens problem which asks you to place 8 queens on a chess board so no two attack each other. In this problem, you will be given locations of queens and knights and pawns and asked to find how many of the unoccupied squares on the board are not under attack from either a queen or a knight (or both). We'll call such squares "safe" squares. Here, pawns will only serve as blockers and have no capturing ability. The board below has 6 safe squares. (The shaded squares are safe.)

▨	**K**		**Q**
▨	▨	**P**	**Q**
	▨		
▨		▨	

Recall that a knight moves to any unoccupied square that is on the opposite corner of a 2 × 3 rectangle from its current position; a queen moves to any square that is visible in any of the eight horizontal, vertical, and diagonal directions from the current position. Note that the movement of a queen can be blocked by another piece, while a knight's movement can not.

Input

There will be multiple test cases. Each test case will consist of 4 lines. The first line will contain two integers n and m, indicating the dimensions of the board, giving rows and columns, respectively. Neither integer will exceed 1000. The next three lines will each be of the form

$k\ r_1\ c_1\ r_2\ c_2\ \cdots\ r_k\ c_k$

indicating the location of the queens, knights and pawns, respectively. The numbering of the rows and columns will start at one. There will be no more than 100 of any one piece. Values of $n = m = 0$ indicate end of input.

Output

Each test case should generate one line of the form
`Board` b `has` s `safe squares.`
where b is the number of the board (starting at one) and you supply the correct value for s.

Sample Input

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

```
1 2 3
2 3
1 1 2
1 1 1
0
1000 1000
1 3 3
0
0
0 0
```

Sample Output

```
Board 1 has 6 safe squares.
Board 2 has 0 safe squares.
Board 3 has 996998 safe squares.
```