Goat Riders

Last night, Lea watched an interesting documentary about some of the more remote areas of the world. Deep in the mountains, there is still one clan left living according to the old ways.

They are called the "Goat Riders" - and they are living in a very harsh environment. Upon reaching maturity, every member of the clan receives a goat as a lifelong companion. Together, they will climb the mountains in search of food and shelter.

Surviving during the winter is especially hard, since sometimes the snow fills entire valleys and everyone that stays there overnight freezes to death. So if the snow gets too deep, the goat riders have to climb to higher ground - otherwise both goat and rider freeze to death.

According to traditions, the clan has divided his territory into grid sectors (indexes with a row and column). Every goat rider can only move one single grid sector per day, since they still need to search for food. Each grid sector is small enough that there is only enough shelter for one single goat rider to survive there.

The goat riders have survived in this harsh environment for centuries and have learned to read the weather. They have learned to perfectly predict how deep the snow will be throughout the winter.

Given a description of the territory, the starting location of the riders and the height of snow on each day until the start of spring, what is the maximum number of goat riders that can survive the winter?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing three integers n, k and d, where n is the size of the goat riders territory (the territory is an $n \times n$ grid), k is the number of goat riders, and d is the number of nights the goat riders have to survive.

n lines follow, each containing n integers $height_i$, representing the height of each grid sector of the territory. The first row in the input is row 0, and the last is row $n-1$. The first column in each row is column 0, and the last is column $n-1$.

k lines follow, each containing two integers r, c, indicating that on day 0 there is a goat rider in row r and column c. No two goat riders will be in the same position.

d lines follow, with the j-th line containing a single integer $level_j$, indicating that during the night from day $(j-1)$ to day j the snow will rise or fall to level $level_j$. (So the first line will contain the level that the snow will be on the morning of day 1.) The goat riders can still move during day 0. A grid section is considered to be *frozen* if its $height \leq level$ of the snow. All goat riders that start their day on a frozen section freeze to death.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1 and x is the maximum possible number of surviving goat riders.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 100$
- $0 \leq k \leq 100$
- $1 \leq d \leq 24$
- $0 \leq height_i \leq 100$
- $0 \leq level_j \leq 100$
- No goat rider can leave the territory.

Sample explanation

In sample 1, case 1, there is only one single goat rider, starting at $(1, 1)$ at height 1. On the first day, the rider moves to $(2, 1)$ with height 4. During the first night, the snow falls to height 0. During the second and third day, the rider stays at $(2, 1)$ at height 4 and thus survives until the morning of day 4.

In case 2, there are 3 riders. The first one can stay where he is for one night, then go a column to the right while the snow is low and then go another one to the right before the snow rises again. The second rider could do the same: stay for the first night, then go up a row, then go up another row. Unfortunately, the first rider is also at the same grid section, so only one of them survives. The third rider cannot reach the safe peaks higher than 6, so he is doomed anyway.

Sample Input 1

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

4 3 3
4 1 9 1
1 6 1 0
1 1 1 2
4 1 1 9
0 0
1 1
3 0
3
0
6
```

Sample Output 1

```
Case #1: 1
Case #2: 1
```

Sample Input 2

```
4
5 2 3
4 1 9 6 3
6 1 8 0 3
5 7 4 2 1
1 4 9 2 2
4 3 5 6 4
1 0
4 2
4
0
6

6 5 5
7 0 5 0 6 8
9 4 3 0 7 8
6 1 8 4 6 4
5 1 2 6 4 2
2 7 7 8 2 1
3 2 2 7 7 5
5 4
5 1
4 1
5 3
3 1
3
7
0
3
6

3 5 3
6 9 5
4 2 0
4 1 5
1 0
2 2
0 0
0 1
2 0
3
7
6

9 4 5
3 0 7 3 0 9 4 2 8
0 5 1 2 4 5 2 5 7
5 0 3 0 6 7 0 9 0
4 2 6 0 7 6 4 3 7
1 3 2 5 2 1 2 3 8
9 5 9 0 1 8 8 9 0
3 6 2 0 5 4 1 9 6
1 8 3 3 3 4 8 0 8
2 9 6 0 0 7 0 4 5
1 2
0 2
1 7
6 3
1
6
6
1
1
```

Sample Output 2

```
Case #1: 2
Case #2: 1
Case #3: 1
Case #4: 3
```