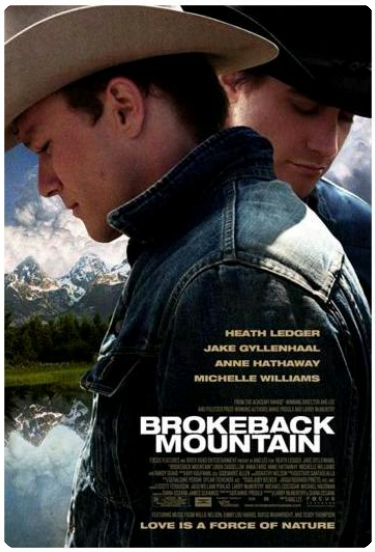
The Movies (1)

Problem Statement



John and Brus are going to a theater to see a very interesting movie. They would like to have seats next to each other in the same row. The theater contains **n** rows, with **m** seats in each row. Rows are numbered 1 to **n** from front to back, and seats are numbered 1 to **m** from left to right. Some of the seats are already reserved, but John and Brus can book any of the available seats. You are given a sequence of **row** and **seat** with size **C**. The i-th elements of **row** and **seat** are the row number and seat number of the i-th reserved seat. All remaining seats are available.
Return the number of ways for John and Brus to book two available seats next to each other in the same row.

Notes

Two bookings are considered different only if one contains a seat that the other does not contain. In other words, they don't need to decide which seat John sits in and which seat Brus sits in.

Input

The input starts with an integer **A** to indicate the number of testcases.

Each testcase starts with two integers, **n** and **m** as in problem statement. On the next line, an integer **C** is given to indicate the size of the reserved seat sequence. On the next line, there are **C** integers to indicate the row number of the reserved seat. Finally, there are C integers to indicate the seat number of the reserved seat

Output

For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1) and y is the number of ways for John and Brus to book two available seats next to each other in the same row.

Constraints

1. **n** and **m** will each be between **1 and 1,000,000,000**, inclusive.
2. **C** will between 1 and 47 elements, inclusive.
3. **row** and **seat** will contain the same number of elements.
4. Each element of **row** will be between 1 and **n**, inclusive.
5. Each element of **seat** will be between 1 and **m**, inclusive.
6. All pairs (**row**[i], **seat**[i]) will be distinct.

Sample I/O

Input	Output
4 2 3 2 1 2 2 3 2 3 6 1 1 1 2 2 2 1 2 3 1 2 3 4 7 1 1 1 10 8 10 1 9 6 10 6 7 9 3 9 2 7 7 3 3 7 1 5 1 6 2	Case #1: 1 Case #2: 0 Case #3: 23 Case #4: 54

Explanation

For input 1:

The first and the second seat in the second row are the only two free seats next to each other in the same row.

For input 2: There are no free seats in the theater.

Hints:

- stl :: map or set are good container
- stl :: set< pair<int,int> > is also good to use