

Mysterious For

Time Limit: 4000/2000 MS (Java/Others) Memory Limit: 65536/65536 K (Java/Others)
Total Submission(s): 654 Accepted Submission(s): 244

Problem Description

MatRush is an ACMer from ZJUT, and he always love to create some special programs. Here we will talk about one of his recent inventions.

This special program was called "Mysterious For", it was written in C++ language, and contain several simple for-loop instructions as many other programs. As an ACMer, you will often write some for-loop instructions like which is listed below when you are taking an ACM contest.

```
for (int i = 0; i < n; i++) {
    for (int j = i; j < n; j++) {
        for (int k = j; k < n; k++) {
            blahblahblah();
        }
    }
}
```

Now, MatRush has designed m for-loop instructions in the "Mysterious For" program, and each for-loop variable was stored in an array a[], whose length is m.

The variable i represents a for-loop instructions is the i-th instruction of the "Mysterious For" program. There only two type of for-loop instructions will occur in MatRush's "Mysterious For" program:

1-type: if a for-loop belongs to 1-type, it will be an instruction like this:

```
for (int a[i] = 0; a[i] < n; a[i]++) {
    ...
}
```

2-type: if a for-loop belongs to 2-type, it will be an instruction like this:

```
for (int a[i] = a[i - 1]; a[i] < n; a[i]++) {
    ...
}
```

In addition, after the deepest for-loop instruction there will be a function called HopeYouCanACIt(), here is what's inside:

```
void HopeYouCanACIt() {
    puts("Bazinga!");
}
```

So, the "Mysterious For" program, obviously, will only print some line of the saying: "Bazinga!", as it designed for.

For example, we can assume that n equals to 3, and if the program has three 1-type for-loop instructions, then it will run $3^3=27$ times of the function HopeYouCanACIt(), so you will get 27 "Bazinga!" in total. But if the program has one 1-type for-loop instruction followed by two 2-type for-loop instructions, then it will run $3+2+1+2+1+1=10$ times of that function, so there will be 10 "Bazinga!" on the screen.

Now MatRush has the loop length n and m loop instructions with certain type, then he want to know how many "Bazinga!" will appear on the screen, can you help him? The answer is too big sometimes, so you just only to tell him the answer mod his QQ number:364875103.

All for-loop instructions are surely nested. Besides, MatRush guaranteed that the first one belongs to the 1-type. That is to say, you can make sure that this program is always valid and finite. There are at most 15 1-type for-loop instructions in each program.

Input

First, there is an integer $T(T \leq 50)$, the number of test cases.

For every case, there are 2 lines.

The first line is two integer $n(1 \leq n \leq 1000000)$ and $m(1 \leq m \leq 100000)$ as described above.

The second line first comes an integer $k(1 \leq k \leq 15)$, represents the number of 1-type loop instructions, then follows k distinctive numbers, each number is the i -th 1-type loop instruction's index(started from 0), you can assume the first one of this k numbers is 0 and all numbers are ascending.

All none 1-type loop instructions of these m one belongs to 2-type.

Output

For each certain "Mysterious For" program, output one line, "Case #T: ans", where T stands for the case number started with 1, and ans is the number of "Bazinga!" mod 364875103.

Sample Input

```
5
3 3
3 0 1 2
3 3
1 0
3 3
2 0 2
4 4
4 0 1 2 3
10 10
10 0 1 2 3 4 5 6 7 8 9
```

Sample Output

```
Case #1: 27
Case #2: 10
Case #3: 18
Case #4: 256
Case #5: 148372219
```

Hint

For the third program, the code is like this:

```
for (int a[0] = 0; a[0] < n; a[0]++) {
    for (int a[1] = a[0]; a[1] < n; a[1]++) {
        for (int a[2] = 0; a[2] < n; a[2]++) {
            HopeYouCanACIt();
        }
    }
}
```

```
}
```

Because $n = 3$, the answer is $3*3+2*3+1*3=18$.

Source

2012 Multi-University Training Contest 8

Recommend

zhuyuanchen520