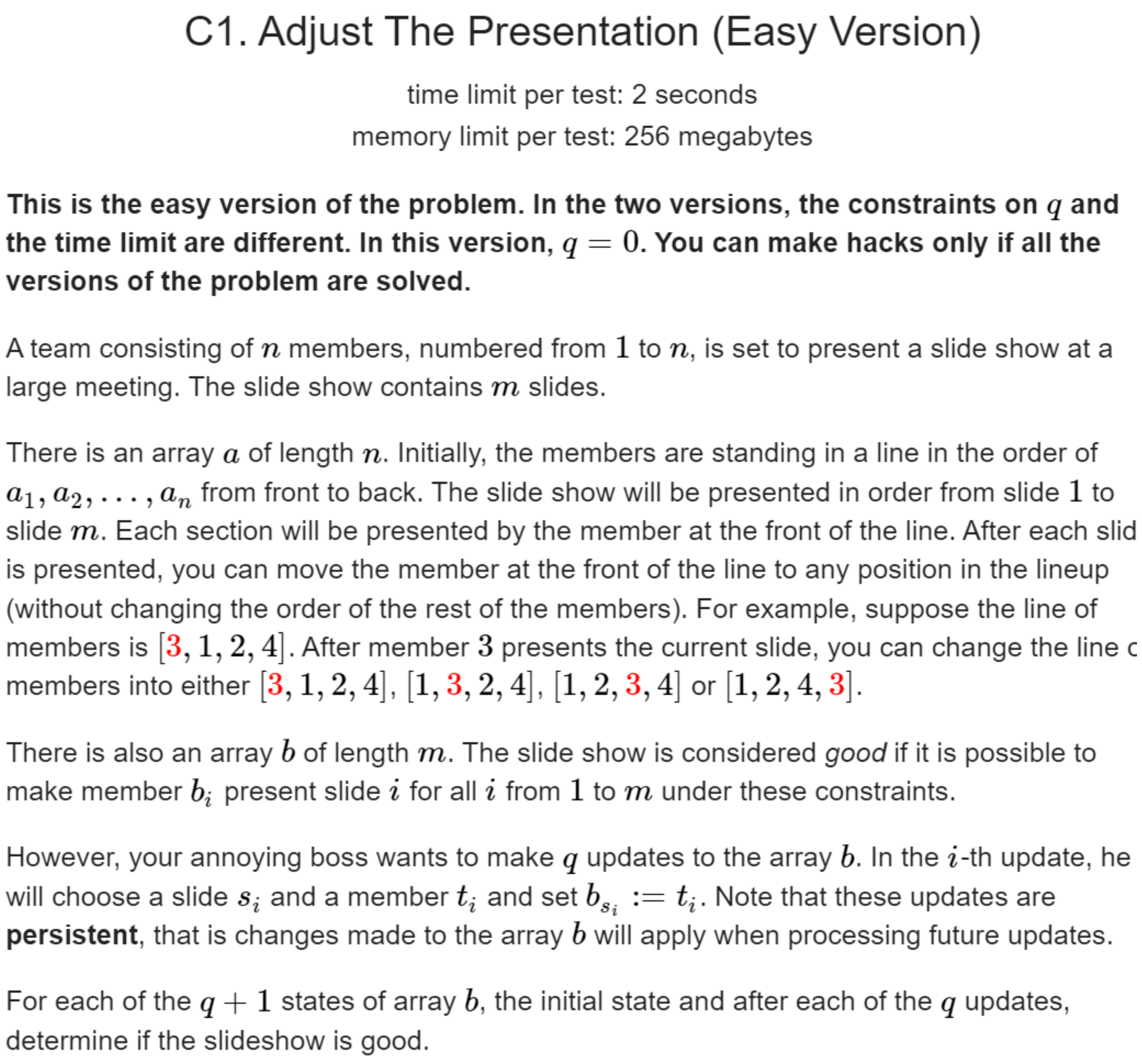
C1. Adjust The Presentation (Easy Version) Zadanie z Codeforces / Div. 2

Zadanie pochodzi z platformy Codeforces:
https://codeforces.com/contest/2021/problem/C1



C1. Adjust The Presentation (Easy Version)

time limit per test: 2 seconds
memory limit per test: 256 megabytes

This is the easy version of the problem. In the two versions, the constraints on q and the time limit are different. In this version, $q = 0$. You can make hacks only if all the versions of the problem are solved.

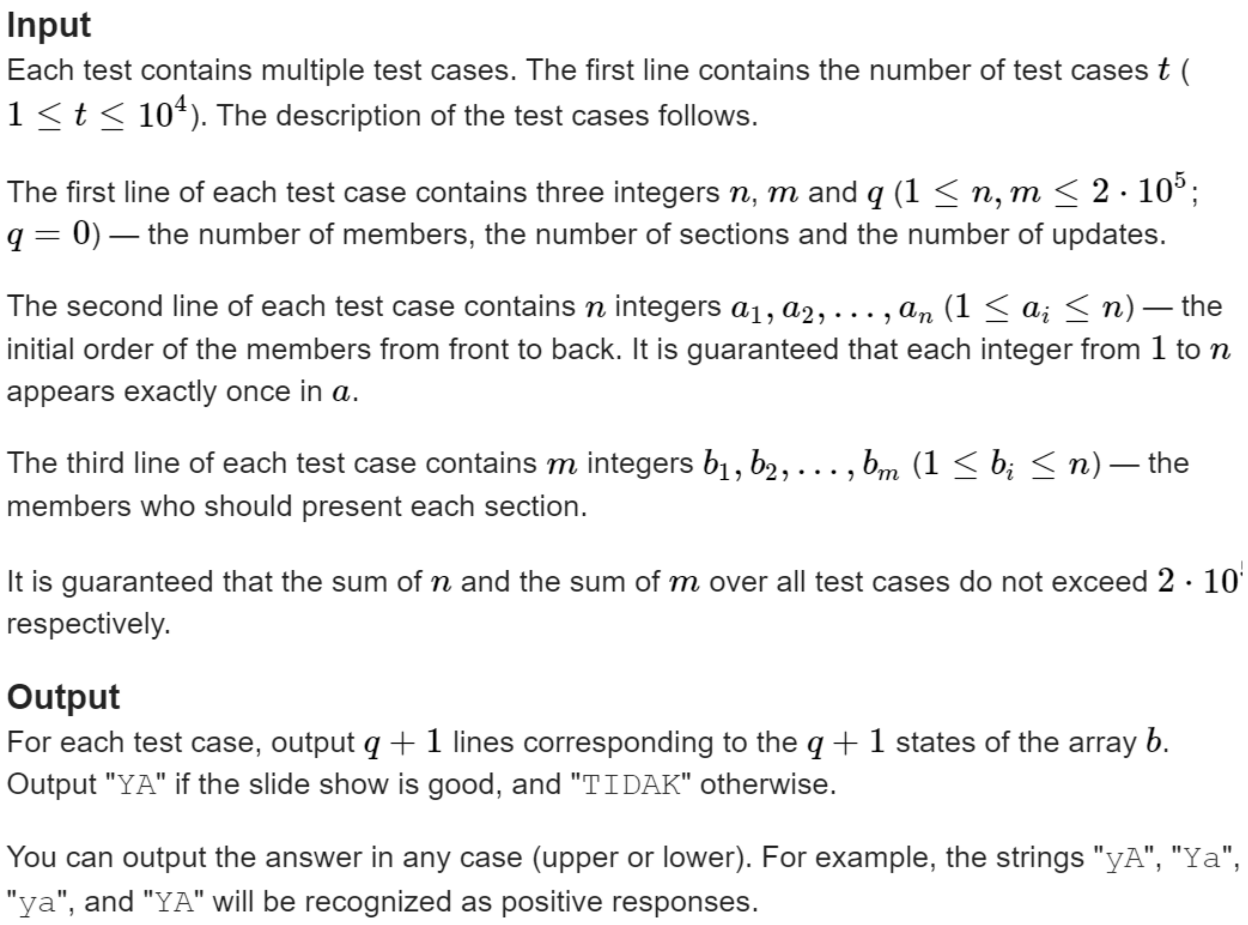
A team consisting of n members, numbered from 1 to n, is set to present a slide show at a large meeting. The slide show contains m slides.

There is an array a of length n. Initially, the members are standing in a line in the order of $a_1, a_2, \ldots, a_n$ from front to back. The slide show will be presented in order from slide 1 to slide m. Each section will be presented by the member at the front of the line. After each slid is presented, you can move the member at the front of the line to any position in the lineup (without changing the order of the rest of the members). For example, suppose the line of members is $[3, 1, 2, 4]$. After member 3 presents the current slide, you can change the line c members into either $[3, 1, 2, 4]$, $[1, 3, 2, 4]$, $[1, 2, 3, 4]$ or $[1, 2, 4, 3]$.

There is also an array b of length m. The slide show is considered *good* if it is possible to make member b_i present slide i for all i from 1 to m under these constraints.

However, your annoying boss wants to make q updates to the array b. In the i-th update, he will choose a slide s_i and a member t_i and set $b_{s_i} := t_i$. Note that these updates are **persistent**, that is changes made to the array b will apply when processing future updates.

For each of the $q + 1$ states of array b, the initial state and after each of the q updates, determine if the slideshow is good.



Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains three integers n, m and q ($1 \le n, m \le 2 \cdot 10^5$; $q = 0$) — the number of members, the number of sections and the number of updates.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$) — the initial order of the members from front to back. It is guaranteed that each integer from 1 to n appears exactly once in a.

The third line of each test case contains m integers $b_1, b_2, \ldots, b_m$ ($1 \le b_i \le n$) — the members who should present each section.

It is guaranteed that the sum of n and the sum of m over all test cases do not exceed $2 \cdot 10$ respectively.

Output

For each test case, output $q + 1$ lines corresponding to the $q + 1$ states of the array b. Output "YA" if the slide show is good, and "TIDAK" otherwise.

You can output the answer in any case (upper or lower). For example, the strings "yA", "Ya", "ya", and "YA" will be recognized as positive responses.

Example 1

Input

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

Output

```
YA
YA
TIDAK
```

Note

For the first test case, you do not need to move the members as both slides are presented by member 1, who is already at the front of the line.

For the second test case, the following is a possible way to move members so that the presentation is good:

1. $[1, 2, 3]$, do not move member 1.
2. $[1, 2, 3]$, move member 1 after member 3.
3. $[2, 3, 1]$, move member 2 after member 3.
4. $[3, 2, 1]$, do not move member 3.
5. $[3, 2, 1]$, move member 3 after member 1.
6. $[2, 1, 3]$, do not move member 2.