1261 – K-SAT Problem

SAT was the first known NP-complete problem. The problem remains NP-complete even if all expressions are written in conjunctive normal form with 3 variables per clause (3-CNF), yielding the 3-SAT problem. A **K-SAT** problem can be described as follows:

There are **n** persons, and **m** objects. Each person makes **K** wishes, for each of these wishes either he wants to take an object or he wants to reject an object. You have to take a subset of the objects such that every person is happy. A person is happy if at least **one** of his **K** wishes is kept. For example, there are 3 persons, 4 objects, and **K = 2**, and

Person 1 says, "take object 1 or reject 2."
Person 2 says, "take object 3 or 4."
Person 3 says, "reject object 3 or 1."

So, if we take object 1 2 3, then it is not a valid solution, since person 3 becomes unhappy. But if we take 1 2 4 then everyone becomes happy. If we take only 4, it's also a valid solution. Now you are given the information about the persons' wishes and the solution we are currently thinking. You have to say whether the solution is correct or not.

Input

Input starts with an integer **T (≤ 100)**, denoting the number of test cases.

Each case starts with a line containing three integers **n**, **m**, **K (1 ≤ n, m, K ≤ 30)**. Each of the next **n** lines contains **K** space separated integers where the **i**th line denotes the wishes of the **i**th person. Each of the integers in a line will be either positive or negative. Positive means the person wants the object in the solution; negative means the person doesn't want that in the solution. You can assume that the absolute value of each of the integers will lie between **1** and **m**.

The next line contains an integer **p (0 ≤ p ≤ m)** denoting the number of integers in the solution, followed by **p** space separated integers each between **1** and **m**, denoting the solution. That means the objects we have taken as solution set.

Output

For each case, print the case number and **'Yes'** if the solution is valid or **'No'** otherwise.

Sample Input	Output for Sample Input
2 3 4 2 +1 -2 +3 +4 -3 -1 1 4 1 5 3 +1 -2 +4 2 2 5	Case 1: Yes Case 2: No