Paradox Sort

Problem

Vlad likes candies. You have a bag of different candies, and you're going to let Vlad keep one of them. You choose an order for the candies, then give them to Vlad one at a time. For each candy Vlad receives (after the first one), he compares the candy he had to the one he was just given, keeps the one he likes more, and throws the other one away.

You would expect that for any order you choose, Vlad will always end up with his favorite candy. But this is not the case! He does not necessarily have a favorite candy. We know for any pair of candies which one he will prefer, but his choices do not necessarily correspond to a simple ranking. He may choose Orange when offered Orange and Lemon, Banana when offered Orange and Banana, and Lemon when offered Lemon and Banana!

There is a particular candy you want Vlad to end up with. Given Vlad's preferences for each pair of candies, determine if there is an ordering such that Vlad will end up with the right candy. If there is, find the lexicographically-smallest such ordering.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case will start with a line containing the integers **N** and **A**, separated by a space. **N** is the number of candies, and **A** is the number of the candy we want Vlad to finish with. The candies are numbered from 0 to **N**-1. The next **N** lines each contains **N** characters. Character j of line i will be 'Y' if Vlad prefers candy i to candy j, 'N' if Vlad prefers candy j to candy i, and '-' if i = j. Note that if i ≠ j, the jth character of the ith row must be different from the ith character of the jth row.

Output

For each test case output "Case #x: ", where x is the case number, followed by either "IMPOSSIBLE" or a space-separated list of the lexicographically-smallest ordering of candies that leaves Vlad with **A**.

Limits

Memory limit: 1 GB.
1 ≤ **T** ≤ 100.

Small dataset

Time limit: 60 seconds.
1 ≤ **N** ≤ 10.

Large dataset

Time limit: 120 seconds.
1 ≤ **N** ≤ 100.

Sample

Sample Input

```
3
2 0
-Y
N-
2 0
-N
Y-
4 3
-YNN
N-YY
YN-Y
YNN-
```

Sample Output

```
Case #1: 0 1
Case #2: IMPOSSIBLE
Case #3: 1 2 0 3
```