

# Possible Friends

Josué, one of the undergraduate students in PUCMM, is developing a Social Network, but he is having difficulties in identifying the person that has more possible friends. Two persons are possible friends if they are not friends and they have at least one common friend, for example if person A is friend only to person B, and person B is friend of C, then, A and C are possible friends. You are about to help him in this task. Given the network table, you have to write a program that finds the person that has more possible friends, if more than one person matches this criteria, then select the one that comes first ( the one that has the lower ID).

## Input

The first line is **T** ( $1 \leq T \leq 1,000$ ), the number of test cases, then **T** test cases follow.

Each test case consists in a 'Y' or 'N' square matrix(**MxM**) representing the friendship of the network, where **M** is the number of persons.

### Constraints

**M** ( $1 \leq M \leq 50$ )

The square matrix has **M** lines, each line has **M** characters.

The first line of the matrix corresponds to person 0, the next line to person 1, and so on.

On each line the leftmost character corresponds to person 0, the next character to person 1, and so on.

So if character *j* of the line *i* is 'Y', it means that person 'i' is a friend of 'j'.

For each person *i*, character *i* of line *i* will be 'N'.

For each *i,j* character *j* of line *i* will be the same as character *i* of line *j*.

## Output

For each test case, you have to output one line containing the ID of the person(0-based) that has more possible friends and the number of possible friends this person has, separated by a blank space.

## Example

### Input:

```
3
NYN
YNY
NYN
NYYY
YNNY
YNNN
YYNN
NNYNNNN
```

NNYNNNN  
YYNYNNN  
NNYNYNN  
NNNYNYY  
NNNNYNN  
NNNNYNN

**Output:**

0 1  
2 2  
3 4