Problem J: Jungle Trail

Time limit: 15s, memory limit: 1GB.

In the mobile game *Jungle Trail*, you are given a rectangular $n \times m$ board divided into $n \cdot m$ squares. Each square is either empty, blocked (impassable) or contains a den of snakes, either poisonous or benign (not poisonous). If a square contains a den of snakes, then either all the snakes on a given field are poisonous, or all are benign.

The game allows you to tap any column or any row of the board. If you tap a column, all poisonous snakes in this column are turned to benign, and vice versa. Similarly, if you tap any row, all snakes in the row change their state. You can tap each row/column only once. If a den is in a tapped row as well as in a tapped column, its state returns to the original one.

After performing all those operations, you must find a trail through the jungle: a path which starts at the top left corner, in every move goes either one square down or one to the right, ends at the bottom right corner and never passes through a den of poisonous snakes or a blocked field.

Input

The first line of input contains the number of test cases z ($1 \leqslant z \leqslant 500$). The descriptions of the test cases follow.

The first line contains two integers n and m ($2 \leqslant n, m \leqslant 2\,000$).

Each of the following n lines contains m characters `.`, `#`, `O` (capital "o") and `@` (at sign), meaning an empty square, blocked square, den of benign snakes and den of poisonous snakes, respectively. You may assume that the top left corner and the bottom right corner are not blocked.

Neither the sum of n values over all test cases nor the sum of all m values exceed 15 000.

Output

For every test case output the solution in the following format:

The first line should contain `TAK` if a jungle trail is possible or `NIE` if it isn't.

If the answer is `TAK`, in the next three lines output:

- A sequence of n characters T or N, the i-th character being T if the i-th row should be tapped, N if not;
- A sequence of m characters T or N, determining in the same way whether the columns should be tapped;
- A sequence of $n + m - 2$ characters P or D denoting the trail: P means a move right, D means a move down.

Example

For an example input:	a possible correct output is:
1 4 5 ..#.. @@O@@ ##@#O ..@.@	TAK NTNN NNTNT DPPDDPP

Explanation

After tapping the rows and columns described on the output, the board is in the following state:

```
..#..
OOOO@
##O#@
..O.O
```

Now the given path goes only through `.` and `O` squares.