844   Pousse

Recently, the organizers of the International Conference on Functional Programming had a programming contest for people to show that their language was best. The contest main web page is at `http://www.ai.mit.edu/extra/icfp-contest/`. Each entrant had to write an implementation of the game "pouse", for which the description went as follows:

Description of the game

The name of the game is "pousse" (which is French for "push"). It is a 2 person game, played on an N by N board (the original game was 4×4 but $N \times N$ is a simple generalisation to adjust the difficulty of the game, and its implementation). Initially the board is empty, and the players take turns inserting one marker of their color ('X' or 'O') on the board. The color 'X' always goes first. The columns and rows are numbered from 1 to N, starting from the top left, as in:

```
   1 2 3 4
  +-+-+-+-+
1 | | | | |
  +-+-+-+-+
2 | | | | |
  +-+-+-+-+
3 | | | | |
  +-+-+-+-+
4 | | | | |
  +-+-+-+-+
```

    A marker can only be inserted on the board by sliding it onto a particular row from the left or from the right, or onto a particular column from the top or from the bottom. So there are $4 * N$ possible "moves" (ways to insert a marker). They will be named 'Li', 'Ri', 'Ti', 'Bi' respectively, where i is the number of the row or column where the insertion takes place.

    When a marker is inserted, there may be a marker on the square where the insertion takes place. In this case, all markers on the insertion row or column from the insertion square upto the first empty square are moved one square further to make room for the inserted marker. Note that the last marker of the row or column will be pushed off the board (and must be removed from play) if there are no empty squares on the insertion row or column.

    A row or a column is a "straight" of a given color, if it contains N markers of the given color.

    The game ends when an insertion creates a configuration with more straights of one color than straights of the other color; the player whose color is dominant (in number of straights) WINS.

Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

    The standard input of the program will contain a number $N \le 100$ on the first line and this will be followed by a sequence of moves (in the notation previously described) with one move per line. There are no intervening spaces or empty lines.

    The program can assume that all moves in the sequence are valid.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

The organizers then want to play one submitted program against the others to see which is best. So they need to know when one program wins.

Your job is to write a program to determine when a game has been won. The input to your program is the same as described above: an initial number followed by a sequence of moves. As soon as a move produces a winning board position, your program should print out whether 'X WINS' or 'O WINS', and exit. If a line containing 'QUIT' is read before a winner is declared, your program should print out 'TIE GAME' and exit. As a failsafe, the last line of every input will be a 'QUIT' line.

Sample Input

```
2

4
L2
T2
L2
B2
R2
QUIT

4
L2
T2
L2
B2
R2
T1
L2
QUIT
```

Sample Output

```
TIE GAME

X WINS
```