Bono v3

Kang and Kung are board games enthusiast. However, they only like to play deterministic games, such as chess. Since there are only a few deterministic games, they decided to create a new one. This game is called 'Bono'.

The rule of Bono is simple. The board consist of a 3x3 grid. The game is turn based with 2 players. On each turn, the current player must fill an empty cell with a piece of water spinach. Players may not move any water spinaches that have been placed. The board will destroy itself if there is a row or column or diagonal consisting of 3 pieces of water spinach. A player loses if he can't make a move in his turn.

Of course, the first player will always win the game if he plays optimally. That is why they created Bono v2. In Bono v2, the number of boards used in a game is N ($N \leq 1000$). On each turn, the current player must fill an empty cell on an undestroyed board with the same ruling as Bono. A player loses if he can't make a move in his turn.

Bono v2 is still too easy since if they both play optimally, player 1 will always win if N is odd, and player 2 will always win if N is even. To solve this matter, Bono v3 is created. In Bono v3, the initial state of each board might not be empty. Some cells might already be occupied with a piece of water spinach. Other than that, the rules are same as Bono v2.

Kang and Kung decide to play Bono v3 T times ($T \leq 1000$). Kang always moves first and they both play optimally. For each game, who will win?

Input

First line of input is T, the number of games ($T \leq 1000$).

For each game, first line is N, the number of boards ($N \leq 1000$). Next N lines consists of the starting boards. A board is represented with a 9 digit binary string. Cell in (r, c) position is represented by (r-1)x3+c th character in the string. 0 means the cell is empty, 1 means the cell is filled.

Output

For each game, output a line containing the winner's name.

Example

Input:
3
1
000000000
2
000000000
000000000
2
100010000
001010000

Output:
Kang
Kung
Kung