

Checkmate

Annu loves playing chess. She is playing a game of chess with Kushagra.
Now Annu is in a very dominating position. She has captured all the chess pieces of Kushagra.
Only the King is left. And Annu have three chess pieces left - King, Queen and Knight.
You know the positions of all the 4 chess pieces.
If the current configuration of the pieces is checkmate for kushagra, then Kushagra had already lost.
else Annu have one last turn to give checkmate to kushagra and defeat him.

You have to predict that Kushagra lost the game or he has survived the last chance.
If Kushagra lost the game print - "CHECKMATE"
else print - "LUCKY"

About [Chess](#)

The left-lowermost square is represented as [0,0] and the right-topmost square as [7,7].

Input

The first input line contains integer T ($T \leq 10$) which represents the number of testcases.
For every testcase we have 8 integers.
 $T_1, T_2, K_1, K_2, Q_1, Q_2, N_1, N_2$. [$0 \leq T_1, T_2, K_1, K_2, Q_1, Q_2, N_1, N_2 \leq 7$].
 T_1, T_2 -> Position of Kushagra's King.
 K_1, K_2 -> Position of Annu's King.
 Q_1, Q_2 -> Position of Annu's Queen.
 N_1, N_2 -> Position of Annu's Knight.

No two chess pieces lie on the same square.

Output

Print "CHECKMATE" if kushagra lost, else "LUCKY".

Sample tests

Input:

```
3
6 6 0 7 3 6 0 4
7 0 6 2 5 0 7 2
3 2 3 0 1 3 4 1
```

Output:

```
LUCKY
CHECKMATE
CHECKMATE
```

In first testcase there is no chance that Annu can give checkmate to Kushagra in one move.

In second testcase Kushagra is already in checkmate position.

In third testcase Annu can give checkmate to Kushagra in one move.

By moving the queen from [1,3] to [3,3].

New checkmate configuration -> 3 2 3 0 3 3 4 1