After he has learned how to play Nim game, Mike begins to try another stone game which seems much easier.

The game goes like this: Two players start the game with a pile of n stones. They take stones from the pile in turn and every time they take at least one stone. The one who goes first can take at most $n-1$ stones for his first move. From then on a player can take at most k times as many stones as his opponent has taken last time. For example, if one player take m stones in his turn, then the other player can take at most $k \times m$ stones next time. The player who takes the last stone wins the game. Suppose that those two players always take the best moves and never make mistakes, your job is to find out who will definitely win the game.

Input

The first line contains a integer t, indicating that there are t test cases following ($t \leq 20$).

Each test case is a line consisting of two integer n and k. ($2 \leq n \leq 10^8$, $1 \leq k \leq 10^5$).

Output

For each test case, output one line starting with '`Case N:`', N is the case number. And then, if the first player can ensure a winning, print the minimum number of stones he should take in his first turn. Otherwise, print '`lose`'. Please note that there is a blank following the colon.

Hint:

When $k = 1$, the first player will definitely lose if the initial amount of stones is in the set $\{2, 4, 8, 16, 32, \ldots\}$. Let's call this kind of set "first-player-lose set".

When $k = 2$, the first-player-lose set is $\{2, 3, 5, 8, 13, 21, 34, 57, \ldots\}$, which happens to be the Fibonacci sequence starting from 2.

Sample Input

```
5
16 1
11 1
32 2
34 2
19 3
```

Sample Output

```
Case 1: lose
Case 2: 1
Case 3: 3
Case 4: lose
Case 5: 4
```