Once upon a time the following puzzle was suggested to pupils on a regional middle school olympiad on mathematics:

- A set of coins consists of 15 coins: 14 coins are valid while a remaining 15-th coin is a false one. All valid coins have one and the same weight while the false coin has a different weight. One valid coin is marked. Is it possible to identify a false coin balancing coins 3 times at most?

A jury member was a trainer of a team of undergraduates for programming contests. So a question on how to put the puzzle for programming arose naturally. Fin ally the problem was formulated as follows:

- A set of coins consists of N coins: $(N-1)$ coins are valid while a remaining N-th coin is a false one. All valid coins have one and the same weight while the false coin has a different weight. One valid coin is marked. Write a program which for every input pair
  - a number N of coins under question,
  - a limit K of balancing

outputs either '`POSSIBLE`' or '`IMPOSSIBLE`' with respect to existence of a strategy to identify the false coin balancing coins K times at most.

Input

The first line of input contains a single integer T that represents a total amount of different pairs (N, K) to process. Every line of next T lines contains two integers N, $2 \leq N \leq 100$ and K, $0 \leq K \leq 100$.

Output

The output file should contain T lines with '`POSSIBLE`' or '`IMPOSSIBLE`' per line.

Sample Input

```
3
6 2
10 2
15 3
```

Sample Output

```
POSSIBLE
IMPOSSIBLE
POSSIBLE
```