

# UCF Local Contest (Final Round) — August 31, 2024

## Fixing the Tournament

*filename:* tourney

*Difficulty Level:* Easy-Medium

*Time Limit:* 1 second

The most common type of tournament is a single elimination tournament that occurs in rounds. In order to run one of these tournaments, the total number of teams initially included must be a perfect power of 2. For example, if a tournament starts with 16 ( $2^4$ ) teams, in the first round there will be 8 matches, resulting in 8 winning teams and 8 losing teams. In the second round, the 8 winning teams from the first round get paired up into four match ups. The process of pairing the winners continues until the final round when two teams remain and play in the championship match.

For the purposes of this problem, assume that:

- the teams are ranked from 1 to  $2^n$ , for a positive integer  $n$
- the best team is ranked 1 and the worst team is ranked  $2^n$  (this is referred to as “one-based ranking”)
- any time a team with a better rank (lower numbered ranking) plays a team with a worse ranking, the better ranked team wins.

The only thing not set for a tournament is which teams are going to play which teams in each round. You've hacked into the tournament software and can choose any set of pairings of the remaining teams for the matches in each round. Naturally, you'd like to figure out the latest round in the tournament your favorite team might make it to!

### The Problem:

Given the number of rounds for a single elimination tournament and your team's one-based ranking, determine the highest round number your team could reach if you are able to create the match ups for each round optimally.

### The Input:

There is only one input line; it contains two integers:  $n$  ( $2 \leq n \leq 20$ ), indicating the number of rounds for the tournament and  $r$  ( $1 \leq r \leq 2^n$ ), indicating your team's one-based ranking.

### The Output:

Print, on a line by itself, the highest number round your team could reach.

(Sample Input/Output on the next page)

**Sample Input****Sample Output**

4 13	3
6 64	1
20 2	20