

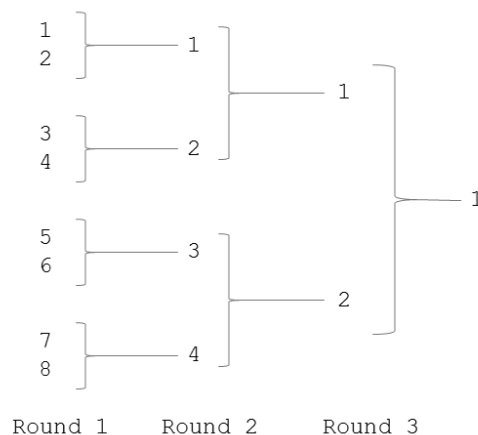
Problem G

Tennis Rounds

Source file name: `tennis.c`, `tennis.cpp` or `tennis.java`

In an N -rounds tennis tournament a group of 2^N players is seeded defining the first round to be played. Seeding means that each player is assigned a number between 1 and 2^N , and this assignment defines the round's draw as it establishes the matches that will be played among the players. In particular, the first round matches are numbered 1, 2, \dots , 2^{N-1} : match k will have player $2\cdot k-1$ vs. player $2\cdot k$, for $1 \leq k \leq 2^{N-1}$.

The winner of a match in the first round advances to the second round and the loser is eliminated. Consequently, the second round has exactly half the players of the first round. Moreover, if the winner of the first round match k would be reassigned the number k , then the second round's draw may be defined exactly as already explained for the first round. This assignment process could be repeated over and over again until there is exactly one player remaining, who happens to be the tournament's champion:



It is clear that the seeding process and the subsequent draws make it possible for any two players to eventually face each other in some round. For example, for $N=3$, players 2 and 5 could play at round 3 (the final), and players 5 and 7 could play at round 2 (one of the semifinals).

The tennis tournament organization is developing an online portal featuring many services. You have been hired to implement one of such services: given the seeding numbers of two players (i.e., their place in the first round ordering), the service should compute the round number in which these two players could eventually have a match against other.

Input

The input consists of several test cases, each one defined by a line containing three blank-separated integers N , i , and j , where N indicates the total number of rounds in the tournament ($1 \leq N \leq 20$), and i , j represent two seeding numbers at the first round ($1 \leq i \leq 2^N$, $1 \leq j \leq 2^N$, $i \neq j$).

The input must be read from standard input.

Output

For each test case, output a line with one integer indicating the round number in which players i and j may have a match in a tournament with N rounds.

The output must be written to standard output.

Sample input	Output for the sample input
3 2 5	3
3 5 7	2
2 1 2	1
2 2 1	1