C: Floor Ceil Fiasco

Time Limit: 1 second

Computers are able to exactly represent base-10 integers up to some limit with a series of 0s and 1s. However, they sometimes struggle with representing base-10 decimals, which have to be rounded off at some point, leading to some weird behavior (for example, try adding 0.1 and 0.2 and printing the result—many languages won't print the mathematically correct answer). To avoid this, many languages will default to integer operations: if both operands are integers, the output is usually an integer as well.

Integer division is problematic if the numerator isn't divisible by the denominator, though. Most languages will round down (floor) the result to the nearest integer, so the integer result may be less than the actual quotient. However, the result from rounding up (ceil) may be greater than the actual quotient.

You will be given two integers n and k. You must divide n by 2 a total of k times. On each division, you may choose to floor the result, rounding it down, or ceil it, rounding it up. Given n and k, output the number of possible resulting integers after applying any combination of k floors or ceils.

Input

The first line of input contains a single integer T ($1 \leq T \leq 50$), representing the number of test cases.

The descriptions of T test cases follow. Each test case consists of a single line containing two integers, n and k ($1 \leq n \leq 10^9$, $0 \leq k \leq 30$).

Output

For each test case, output the number of possible resulting integers after applying any combination of k floors or ceils.

Sample Input 1	Sample Output 1
3 16 3 4 3 7 0	1 2 1

This page is intentionally left blank.