

Problem 2—Multidimensional Tic-Tac-Toe

Mr. Spock is an expert in chess, but he hates the boring, stupid, easy chess that they play on Earth. He plays three-dimensional chess. He would like you to learn to play, too, but realizes you're probably not ready yet, so, instead, you will learn to play multidimensional tic-tac-toe. This game is played on an $n \times n \times n \times \dots \times n$ board. The first player plays X in an empty square, and the second plays O, back and forth, until someone has n of their characters in a row or until the board is filled. The only way to win is to get n of your symbols in a row, orthogonally or diagonally. Notice that even the word “diagonally” can be interpreted in more than one way. On a $3 \times 3 \times 3$ board, you can get three in a row diagonally while staying on the same plane (there are 18 ways to do this), but you can also get three in a row diagonally by hitting every plane (there are 4 ways to do this), plus you can get three in a row within a single row (there are 27 ways to do this), for a total of 49 different ways to win.

INPUT SPECIFICATION. You will be given a set of input cases. Each will consist of two unsigned decimal integers separated by one space and followed by `<EOLN>`. The first represents the number of dimensions of the board. The second represents the number of squares along each dimension. Both numbers will be no smaller than two. “0 0<EOLN>” will follow the last data case. It is not to be processed; it just represents the end of input.

OUTPUT SPECIFICATION. The output cases should appear in the same order as the input cases. Each output case will be of the form “Case c : There are w ways to win.” c is the number of the case, and w is the number of ways to win. Each output case is followed by two `<EOLN>`'s.

SAMPLE INPUT.

```
2 3<EOLN>
3 3<EOLN>
0 0<EOLN>
<EOF>
```

SAMPLE OUTPUT.

```
Case 1: There are 8 ways to win.<EOLN>
<EOLN>
Case 2: There are 49 ways to win.<EOLN>
<EOLN>
<EOF>
```