Problem 4—Triominoes

Even regular dominoes are trite for our brain-obsessed zombie friends. They prefer triominoes, equilateral triangles with positive integers along each edge. Every set of order-*n* triominoes contains exactly one example of each combination of integers from 1 to *n*. Triominoes are said to be unique up to rotation, meaning that the 1-2-3 triomino is the same triomino as the 2-3-1 triomino since you just have to spin one a little to get the other, so only one of these would appear in the set. However, a reflected triomino is not necessarily the same as the original. For example, 1-2-3 is not the same as 3-2-1, since you would have to flip the triomino over to get the 3-2-1, and this would be illegal. So, the 1-2-3 and the 3-2-1 triominoes are both included in the order-3 set. In fact, the total number of triominoes in the order-3 set is 11: {1-1-1, 1-1-2, 1-1-3, 1-2-2, 1-2-3, 1-3-2, 1-3-3, 2-2-2, 2-2-3, 2-3-3, 3-3-3}. Given an integer *n*, you are to determine how many triominoes exist in the set of order *n*.

INPUT SPECIFICATION. You will be given a set of input cases, each of which will begin with an unsigned positive decimal integer $n \leq 100$, representing the size of the cube. The last integer in the file will be 0, which isn't to be processed; it just signifies the end of input. There may be any number of spaces and/or **<EOLN>**'s before, after, and between the integers in the file.

OUTPUT SPECIFICATION. The output cases should appear in the same order as the input cases. Each output case will be of the form demonstrated below. Each output case should be of the form "Case *c*: There are *t* triominoes numbered to *n*.**<EOLN>**" where *c* is the input case, *t* is the number of triominoes in the set of order *n*, *n* is the input case.

SAMPLE INPUT.

```
3•4<EOLN>
0<EOLN>
<EOF>
```

SAMPLE OUTPUT.

```
Case•1:••There•are•11•triominoes•numbered•to•3.<EOLN>
Case•2:••There•are•24•triominoes•numbered•to•4.<EOLN>
<EOF>
```