Problem 3—Rubik Analysis

After a hard night of roaming the street looking for brains, most zombies like to relax with a Rubik's cube. However, the standard 3x3x3 cube is a dull activity, fit only for normal mortals, so they like to go even further. Like most cubists, they are aware that a standard 3x3x3 cube has 8 corner cubes, 12 edge cubes, 6 face cubes, and 1 invisible interior cube. And no matter how much you twist and turn the thing, corner cubes remain corner cubes. Edge cubes remain edge cubes, etc. The 4x4x4 cube has 8 corner cubes, 24 edge cubes, 24 face cubes, and 8 internal cubes, and again each cube never changes roles.

But with the 5x5x5 cube, it starts to get more interesting. It has 8 corner cubes, certainly. But its edge cubes fall into two classifications: 24 of them are adjacent to a corner and 12 of them are in the middle of the edge. The face cubes are further divided: 24 are corners of a face section (there are six faces and each has four corners); 24 are edges of a face; 6 are in the center of a face. And the invisible internal cubes are also subdivided, with 8 on the corners of the internal cube, 12 on the edges, 6 on the faces, and 1 in the absolute dead center, for a total of 125. And, again, a cube in each classification cannot move to another group no matter how much you play with that cube.

We can numerically classify the types of cubes: Corners are 0; edges are 1; faces are 2; internals are 3. However, whenever we have an edge, face, or internal cube (anything other than a corner) we classify it further. And if that further classification is itself anything other than a corner, we classify it still further, until we reach a point where the cube is a "local corner," for lack of a better term.

For example, in the 5x5x5 cubes, 8 are classified as 0: the corners. 24 are classified as 1-0: These are the edge cubes adjacent to corners. They are edge cubes, hence the 1-classification, but these cubes are endpoints or "corners" of the edge they're on, so they get followed with 0. The 12 cubes on the middles of their edges are classified as 1-1-0. They are edge cubes, so they get a 1. However, since they are still edge cubes within their own edge, they get a second 1 classification. Finally, since they are endpoints of the "edge within the edge," they end up with a 0 in their classification. 24 cubes are classified as 2-0, corners of the face. 24 are 2-1-0, edges of the face. 6 are 2-2-0, centers of the face. 8 are 3-0, the interior corners. 12 are 3-1-0, the interior edges. 6 are 3-2-0, the interior faces, and 1 is 3-3-0, the dead center interior cube. Notice that ALL classifications end with 0.

Another way to look at is this: The first number tells you what remains after most of the cube has been stripped off., and the subsequent numbers apply to the part that remains. A 3 indicates that the entire outer shell of the cube has been removed, so that we're now looking at a smaller cube. A 2 indicates that what remains is a face of the cube (stripped of outer edges and corners). A 1 indicates that an edge is what remains, also stripped of the corners on either end of the edge. A 0 indicates that a corner—taken alone—is all that remains. For example, on a 7x7x7 cube, the classification 2-1-1-0: The 2 would mean look at the 5x5 square on the cube's face. The first 1 would indicate the edge of length 3 on the 5x5 square. The second 1 would indicate the edge of 1 square within that edge of three squares, and the 0 indicates that last cube taken alone. So, on a 7x7x7 cube, 2-1-1-0 would represent those face cubes, adjacent to an edge, that were in the middle of that edge. There are 24 of that kind of cube in a 7x7x7 model.

Given a size of a Rubik's cube, you are to print out the classifications of cubes contained within and the number of cubes within each classification.

INPUT SPECIFICATION. You will be given a set of input cases, each of which will begin with an unsigned positive decimal integer $n \leq 50$, 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. Note that the classifications appear in lexicographic order. Be sure that you match the format below exactly. Note, in particular, that an extra **<EOLN>** follows each output case.

SAMPLE INPUT.

```
2•6•7<EOLN>
0<EOLN>
<EOF>
```

SAMPLE OUTPUT.

```
Case•1:••2x2x2<EOLN>
-----<EOLN>
0:••8<EOLN>
-----<EOLN>
Total:••8<EOLN>
<EOLN>
Case•2:••6x6x6<EOLN>
-----<EOLN>
0:••8<EOLN>
1-0:••24<EOLN>
1-1-0:••24<EOLN>
2-0:••24<EOLN>
2-1-0:••48<EOLN>
2-2-0:••24<EOLN>
3-0:••8<EOLN>
3-1-0:••24<EOLN>
3-2-0:••24<EOLN>
3-3-0:••8<EOLN>
-----<EOLN>
Total:••216<EOLN>
<EOLN>
Case•3:••7x7x7<EOLN>
-----<EOLN>
0:••8<EOLN>
1-0:••24<EOLN>
1-1-0:••24<EOLN>
1-1-1-0:••12<EOLN>
2-0:••24<EOLN>
2-1-0:••48<EOLN>
2-1-1-0:••24<EOLN>
2-2-0:••24<EOLN>
2-2-1-0:••24<EOLN>
2-2-2-0:••6<EOLN>
3-0:••8<EOLN>
3-1-0:••24<EOLN>
3-1-1-0:••12<EOLN>
3-2-0:••24<EOLN>
3-2-1-0:••24<EOLN>
3-2-2-0:••6<EOLN>
3-3-0:••8<EOLN>
3-3-1-0:••12<EOLN>
3-3-2-0:••6<EOLN>
3-3-3-0:••1<EOLN>
-----<EOLN>
Total:••343<EOLN>
<EOLN>
<EOF>
```