Problem 6—Sweet 16

Hikaru Sulu is an avid fencer. In fact, back in his days at Starfleet Academy, he represented his school on the fencing team, eventually competing in the FCAA Sweet Sixteen competition.

The way it works is that the top 16 teams (and it doesn't have to be 16, but it does have to be a power of 2), are ranked in skill from 1 to 16. In the first round, team 1 plays team 16, team 2 plays team 15, and so forth, the reason being that if strong teams play weak teams, the weak teams will be eliminated first, so the next round will consist of the strong teams. (Of course, upsets can and do happen.) In the next round, team 1 plays team 8, team 2 plays team 7, and so forth. (And if, by some miracle, a weaker team beats a stronger team, it absorbs the rank number of the team it beat. If team 16 beats team 1, it becomes team 1, for example.) In the next round, team 1 plays team 4 and team 2 plays team 3. Then, in the final round, team 1 plays team 2, which is the game everyone is really waiting for anyway.

The trick, and the assignment, is to list the teams in such a way so that a "tree" may easily be drawn to keep track of the rounds. For 16 teams for example, you would list them: 1 16 8 9 4 13 5 12 2 15 7 10 3 14 6 11. Your table would look something like:

```
1  16  8  9  4  13  5  12  2  15  7  10  3  14  6  11
  1      8     4      5      2      7      3      6
     1           4           2           3
        1                       2
                    1
```

The idea is to choose the initial ordering so that at each level, the first team listed in each pair is the stronger team. It turns out there is only one way to do this.

INPUT SPECIFICATION. Each input case will be a positive integer, always a power of 2. There may be any of number of spaces or <**EOLN**>'s before, after, or between the input cases. The last integer in the file will be 0; this is not to be processed; it merely signifies the end of input.

OUTPUT SPECIFICATION. The output cases should appear in the same order as the input cases. Each output case should be in the form "Case *c*: *p*" where *c* is the case number and *p* is the permutation of ranks as described above. There should be one space between adjacent integers on the list. Each output case should be followed by two <**EOLN**>'s.

SAMPLE INPUT.

```
8·16<EOLN>
0<EOLN>
<EOF>
```

SAMPLE·OUTPUT.

```
Case·1:·1·8·4·5·2·7·3·6<EOLN>
<EOLN>
Case·2:·1·16·8·9·4·13·5·12·2·15·7·10·3·14·6·11<EOLN>
<EOLN>
<EOF>
```