Game of Life   P24077_en

Setè Concurs de Programacio de la UPC - Semifinal (2009-06-29)

The game of life is played on an infinite two-dimensional orthogonal grid where the cells are either alive or dead. The rules of the game are: Every alive cell will remain alive in the next time step if and only if exactly two or three of its eight neighbors are alive. A dead cell will get alive in the next time step if and only if it has exactly three neighbors alive.

From this simple set of rules, very complex patterns can occur, which can even simulate Turing machines. Here we are looking for gliders, i.e., patterns that move through the grid either horizontally, vertically or diagonally. Such patterns reappear after a certain number of time steps (called period) displaced several cells in some direction. Note that gliders may move more than one cell on each period (say, three cells to the right after four time steps).

Input

Input consists of several strings, each indicating one of eight possible directions: "N", "NE", "E", "SE", "S", "SW", "W" or "NW".

Output

For every direction, print the smallest pattern (as defined below) that is a glider and moves in that direction. For instance, moving in "NW" direction means that the pattern will reappear displaced x cells to the north and x cells to the west, for some $x \geq 1$. Print an 'X' for the cells that are alive, and a dot for the cells that are dead. Print only the bounding box of the alive cells. That is, the first row, the last row, the first column and the last column must have at least one alive cell. Print an empty line between the output for two consecutive strings.

The smallest pattern is defined as that with the smallest bounding box area. In case of a tie, choose the one with smaller height. In case of a second tie, choose the one lexicographically smaller when reading the rows consecutively from top to bottom, considering that an 'X' goes before a dot. In case that no solution exists with a bounding box area at most 20, or with a period at most four, print "TOO COMPLEX FOR ME" in a line.

Sample input

```
NW
NW
```

Sample output

```
XXX
X..
.X.

XXX
X..
.X.
```

Problem information

Author : Ricardo Martín
Generation : 2024-04-30 17:23:58