Problem Set 6

OUT: 25 OCTOBER, 1994
DUE: 3 NOVEMBER, 1994

A game of tick-tack-toe can be recalled by naming in sequence the moves made by the two players, X and O. If we label the board according to,

1	2	3
4	5	6
7	8	9

then an example of a game would be,

$$X1, O4, X5, O9, X2, O3, X8,$$

This is a win for X,

X	X	O
O	X	
	X	O

The totality of all possible games of tick-tack-toe can be visualized in a tree. A token is placed on the root of the tree and X and O take turns moving the token from its current node to one of the node's children. If the root is said to be at depth 0, and a node other than the root is said to be at depth i if its parent is at depth $i-1$, then moves from even to odd depth nodes are the choice of X, while moves from odd to even depth nodes are the choice of O. A depth i node has $9-i$ children, as there are only $9-i$ unmarked spaces on the game board after i moves by players.

Imagine the game tree fully built. Color blue nodes which contain a win for X, red nodes which contain a win for O, and green nodes which contain a draw. The remaining nodes are uncolored. All colored nodes, and only the colored nodes, are leaves: one plays until a win or draw, and at a win or draw, one plays no further. The point of tick-tack-toe is for X to navigate the token towards a blue node, while O navigates the token towards a red node.

If perfect play is assumed, the coloring of the leaves can be propagated to a coloring of all the nodes. The rule is:

> *If the node is governed by a choice by X, color the node blue if any child is blue, green if no child is blue but some child is green, and red otherwise. If the node is governed by a choice of Y, replace blue by red and red by blue in the previous sentence.*

Proceeding in this way, the root of the tick-tack-toe game tree eventually gets colored green. This reflects the well known fact that tick-tack-toe, when perfectly played by both players, always results in a draw.

The coloring algorithm is the basis for an algorithm $\mathcal{A}$ by which the computer can play a perfect game of tick-tack-toe. The algorithm $\mathcal{A}$ returns blue, red

or green depending upon whether the node is X's choice or O's choice, and depending upon the color returned by a recursive application of $\mathcal{A}$ to some or all of the children, reversing the sense of the choice. Said formally, the algorithm consists of statements of the form,

$$\mathcal{A}(X, n) = \text{Red} \Leftrightarrow \exists n', n' \text{ is a child of } n, \text{ and } \mathcal{A}(O, n') = \text{Red},$$

and so on.

Write a pascal program which plays excellent tick-tack-toe. You may use algorithm $\mathcal{A}$, or come up with your own. The algorithms will be run against each other, in a single elimination competition, to find a winner. Therefore, please use the input/output format illustrated by the following sample run.

```
Welcome to tick-tack-toe by Burt Rosenberg
Would you like to play X or O? O
X> 1
       X |   |
      ---+---+---
         |   |
      ---+---+---
         |   |
O? 4
X> 5
       X |   |
      ---+---+---
       O | X |
      ---+---+---
         |   |
O?
```

and so on. Programs which break during play are considered losers. Losing programs are graded on a scale of 1 to 4. Winning programs are awarded 5 points. Good luck!