**COMP 8901 – Assignment #5 Due date: Nov 26 (11:59pm)**

**Connect 4**

**The Game**

Connect Four is a two-player game played on a grid that is 7 columns wide and 6 rows high. Each player starts with 21 tokens. The RED player begins play by dropping a single red token into the column of his or her choice. When a token is dropped into a column, it falls into the lowest unoccupied position in that column. RED and BLACK alternate moves until a winning configuration appears in the grid, or until the grid has been completely filled with tokens (resulting in a draw). A player wins when the grid contains at least four of his or her tokens in a single row, column, or diagonal.

**Your task**

Write a program that uses minimax search to play a game of Connect Four against an interactive human opponent.

**Requirements**

Your program must do the following:

- Prompt the user to see who will play first (i.e. who is RED).

- If the computer is RED, then it should make a move and then display the board position.

- The player should be prompted to enter the column they want to play in, and the result should be displayed.

- The player and computer should continue alternating moves until the game finishes.

Your program should recognize when the game is over and not permit illegal moves by the player. I am not worried about graphics... you can have a GUI for the game, or you can use a text-based board representation if you prefer.

Internally, you should use a minimax search to determine moves for the computer. This requires a suitable static evaluation function and a search depth. You need to write your own code to implement these.

**Submission**

You need to submit all code, as well as a 1-2 page document that explains the following:

1. How to run your program.
2. The evaluation function that you used, as well as the search depth.
3. And explanation of your algorithm… i.e. is it straight minimax? What kind of pruning does it use? Do you use any other enhancements?

**Evaluation**

You must use minimax to solve this game, with suitable pruning and other enhancements to improve play. Time permitting – we will hold an in-class tournament between game players. The structure of the tournament will be determined later... but it is worth knowing that there will be a competition. You do not need to write your program to interact with other programs... we will run the games manually between programs.

Evaluation will include a performance component. Games that win often will be judged favourably. Programs that lose badly with poor decision making will receive a deduction. Programs that take an inordinate amount of time will also receive a deduction.