CS420 -- Project 3: Game (4-in-a-line)   **Due: 3/06/2018 by 11:59pm**

You are only allowed to use Java, C++

The Game:
You have a 8x8 board, players take turns placing a piece on any grid. First player to get 4 in a line (either a row, or a column; diagonals are NOT counted) wins. The maximum amount of time allowed for generating the next move is 30 seconds. At the very beginning, your program should ask the user for the maximum amount time (in seconds) allowed for the computer to generate the answer and make sure that your search will stop when there's no time left. Your program should also ask the user to decide who is going to move first.

In order to make the process smooth, you are **REQUIRED** to use the following standard in the movement that your program will generate:

```
    1 2 3 4 5 6 7 8        Player vs. Opponent
 A  - - - - - - - - -         1.  e5 d5
 B  - - O - - - - - -         2.  e4 e3
 C  - - O - - - - - -         3.  f4 d4
 D  - X O O O X - - -         4.  e6 e7
 E  - - O X X X O - -         5.  g4 h4
 F  - - X X - - - - -         6.  d6 d3
 G  - - - X - - - - -         7.  d2 c3
 H  - - - O - - - - -         8.  f3 b3 wins
```

There is a sample for Windows executable written in C available with this assignment.

Strategies and Other Requirements

1. To get a full credit, your program must satisfy the requirements listed above, and MUST use alpha-beta pruning to determine the computer's move. Note that, you will need to make some changes to the alpha-beta pruning introduced in the book. These changes include: an evaluation function so that you can evaluate non-terminal states and a cut-off test to replace the terminal test so that your program will return the best solution found so far given a specific period of time.

2. In general, you are allowed 5 seconds to run your algorithm, which should search at least 5 plies deep. If you use languages other than C or C++, the number of plies may be smaller. You can implement iterative deepening search with alpha-beta pruning so that your program will search as deep as possible given the fixed amount of time.

3. Given the same amount of time, if your terminal evaluation function is simple, then your program should be able to search deeper than one with a complex (but maybe better) terminal evaluation function. You can decide whether you want a deeper search and a simple evaluation, or a shallower search with a complex evaluation.

4. Your program is one player, and will attempt to defeat the opponent.

5. Your program must prompt the user for who goes first; player or opponent.

6. The program should detect when an illegal move has been entered and requires the human to re-enter a valid move, for example, trying to place on a non-empty space, or out of bounds

Note: A group of 2 is allowed to work on this project. You can also work individually. Your final executable program should be able to run on the computers in the CS labs. The top team will receive 6 points added to your final grade. The two runners-up will receive 3 points added to your final grade.

How do two programs interact?
To make the interaction interface easy, your program should output what's the current choice for each step. Each program will run on one computer and the final result will be the opposite of each other. Please see the following illustration:

Program running on computer 1 goes first: the program should search for its first move, for example, "e5"

```
     1 2 3 4 5 6 7 8         Player vs. Opponent
  A - - - - - - - - -             1. e5
  B - - - - - - - - -
  C - - - - - - - - -
  D - - - - X - - - -
  E - - - - - - - - -
  F - - - - - - - - -
  G - - - - - - - - -
  H - - - - - - - - -


     Player's move is: e5


   Choose Player's next move:
```

Program running on computer 2: this program is waiting for the first player's move (opponent) to be entered. After which, this computer will search for the best responding move and draw the resulting board.

```
   Choose Opponent's next move: e5

     1 2 3 4 5 6 7 8         Opponent vs. Player
  A - - - - - - - - -            1. e5   d5
  B - - - - - - - - -
  C - - - - - - - - -
  D - - - - 0 - - - -
  E - - - - X - - - -
  F - - - - - - - - -
  G - - - - - - - - -
  H - - - - - - - - -


     Player's move is: d5


     Choose Opponent's next move:
```

This process will continue until the end of the game. The program must announce a win, lose, or draw then terminate. If the opponent goes first the order in which the label and move log appears will have to be changed; e.g., "Opponent vs. Player" or "Player vs. Opponent"

How is the tournament run?

It depends on the number of groups who are willing to participate. Please let me know your decision on participation by 11:59 pm of November 25th.

Student teams will be arranged into groups of 3 teams. The teams in each group will play against each other. Each match contains two games, with each team moving first. A group will play a total of 6 games. A win counts as 1 point, a draw as .5, and a loss as 0 points. Within each group, whoever has the most points is the winner of the group.

The next round will consist of winner teams from the first round and some 2nd place teams. This process will continue.

What to submit?

Similar submission instructions as project 2. With a report on your strategies in generating the evaluation function and implementation of the alpha-beta pruning algorithm.