CSCI 5134 Programming Assignment 1

DUE: Sunday September 27, 2015 at midnight (end of the day)

The game Connect 4 is a two-player board game consisting of 6 rows and 7 columns. The game is also known as four in a row. The object of the game isthat each player takes a turn inserting his/her own chip in a column. The chipshould land in the lowest possible location in the column. For example if thecolumn was empty then the chip will land in the first row of the column andif the column has 3 chips in it, then the new chip will land in the fourthrow of the column and so on.

Each player has a different color set of chips (let's say yellow and red, orblue and red). Players will contend for the board to insert their chips in a random column. Thewinner of the game is the player that is able to connect 4 of his/her samecolor chips in a row either vertically or diagonally. If theboard is filled and no player was able to connect 4 in a row, then the game

will end in a tie.

In this assignment you are to implement a modified version of this game as a multithreaded application that creates three threads, one thread for a referee, and two others to play theconnect 4 game. The only problem here is that threads **do not take turns**, butrather **contend** for the board. This means that a player can go two times in arow if that player can access the board game before the other. It is OK forthis game to be unfair if taking turns doesn't happen.

This is a different version of the producer consumer problem where the producerproduces a game board (6x7) and a single integer indicating if a winner hasbeen found or not. The scenario of running the program goes as follows:

1. The producer produces an empty game board, and indicates that no winnerhas been assigned and play may resume.

2. A thread will only access the board game if it has been released by the producer and also if no winner has been announced and the play may resume.

3. After a player makes a move (based on generating a random number between 1and 7), the player thread should indicate to the referee that it's timeto check for a winner.

\* The player should print the game board after making the move and insert an appropriate pause so the user has enough time to see the board changing.

4. The referee thread will check to see if there's a winner, and if so, itwill announce it and end the game. Otherwise, it will indicate that no winner has been found and play may resume.

\* The referee should either announce a winner or indicate, no winner foundand play should resume.

5. Once play resumes and no winner has been found, the next available player thread may take control, insert a chip, and once again indicate to thereferee it's time to check.

\* The player should print the game board after making the move

STEPS 4 and 5 are repeated until a winner is announced or the game ends in atie.

Your program should make use of mutexes and condition variables to check the availabilty of the board and to check if play may resume for the playerthreads or referee may check the results for the referee thread.

**Schedule an appointment to show a demo to T.A. the week after (or before if you finish early).**

Best of Luck

If you need help with your Unix account:

[http://sceweb.sce.uhcl.edu/support/](https://webmail.uhcl.edu/owa/redir.aspx?SURL=jZ8XNMBUpV1UqhljosYf9q50lEyKrKIvZSXl7n33-ElradLIQb3SCGgAdAB0AHAAOgAvAC8AcwBjAGUAdwBlAGIALgBzAGMAZQAuAHUAaABjAGwALgBlAGQAdQAvAHMAdQBwAHAAbwByAHQALwA.&URL=http%3a%2f%2fsceweb.sce.uhcl.edu%2fsupport%2f)