```
/*****************************************************************************
 * PROGRAMMER : Ali Eshghi
 * STUDENT ID : 1112261
 * CLASS      : CS1B
 * SECTION    : MW 7:30pm
 * Assign #2  : tic-tac-toe game (multi-dimensional arrays)
 * DUE DATE   : 19 September 2019
 *****************************************************************************/

#ifndef MYHEADER_H_
#define MYHEADER_H_

#include<iostream>
#include<iomanip>
#include<string>
#include<stdlib.h>
#include<time.h>
#include<curses.h>
#include<cstdlib>
using namespace std;

/*************
 * VARIABLES *
 *************/

const int ROW_SIZE = 3; //PROCESS - error checking the input for row
const int COL_SIZE = 3; //PROCESS - error checking the input for col


/*****************************************************************************
 * PrintHeader
 *      This function outputs the header into the screen.
 *****************************************************************************/
void PrintHeader(const string MY_NAME,
           const string CLASS,
           const string CLASS_TIME,
           const int    ASSIGN_NUM,
           const string ASSIGN_NAME);

/*****************************************************************************
 * OutputInstruct
 *    This function outputs instructions to the users. There are no input
 *    or output parameters for this function as it only displays text to
 *    the screen.
 *
 *
 *    RETURNS: nothing
 *     Displays the instructions to the user
 *****************************************************************************/
void OutputInstruct();

/*****************************************************************************
 * InitBoard
 * This function initializes each spot in the board to a space ' '.
 *
```

```
 * RETURNS: Board initialized with all spaces
 ******************************************************************************/
void InitBoard(char boardAr[][3]);   // OUT -tic tac toe board
                                     // Done

/******************************************************************************
 * DisplayBoard
 * This function outputs the tic-tac-toe board including the tokens
 * played in the proper format (as described below).
 *
 *          1       2       3
 *       [1][1] | [1][2] | [1][3]
 *              |        |
 * 1            |        |
 *              |        |
 *      -------------------------
 *       [2][1] | [2][2] | [2][3]
 *              |        |
 * 2            |        |
 *              |        |
 *      -------------------------
 *       [3][1] | [3][2] | [3][3]
 *              |        |
 * 3            |        |
 *              |        |
 *
 * * RETURNS: nothing
 *  outputs the current state of the board
 ******************************************************************************/
void DisplayBoard(const char boardAr[][3]); // IN -tic tac toe board
                                            // Done

/******************************************************************************
 * GetPlayers
 *  This function prompts the user and gets the input for the players' names.
 *  playerXwill always contain the name of the player that is using the X token.
 *  playerOwill always contain the name of the player that is using the O token.
 *
 * RETURNS: the players names through the variables playerX and playerO.
 ******************************************************************************/
void GetPlayers(string &playerX,    //OUT -player X's name
                string &playerO,//OUT -player O'x name
                char   &compToken,
                char   &tokenChoice,
                int    option);
                //Done



/******************************************************************************
 * GetAndCheckInp
 *  This functions gets each player's play and checks if the inputed numbers are
 *  in the domain of the row and column of the game. also it checks if the
 *  row and column in the board is empty or no
```

```
111 *
112 * RETURNS: nothing
113 *          it puts the players token in the boardAr
114 ********************************************************************************/
115 void GetAndCheckInp(char    boardAr[][3],char   token,
116                     string  playerX     ,string playerO,
117                     int option      ,char   tokenChoice,
118                     charcompToken);
119                     //Done
120
121 /*******************************************************************************
122  * SwitchToken
123  *   This function switches the active player.
124  *   It takes in a parameter representing the current player's token
125  *   as a character value (either an X or an O) and returns the opposite.
126  *   For example, if this function receives an X it returns an O. If it
127  *   receives and O it returns and X.
128  *
129  * RETURNS: the token opposite of the one in which it receives.
130  ********************************************************************************/
131 char SwitchToken(char token);  // IN -current player'stoken ('X' or 'O')
132                                // Done
133
134
135 /*******************************************************************************
136  *  CheckWin
137  *   This function checks to see if either player has won. Once it is
138  *   possible for a win condition to exist, this should run after each a
139  *   player makes a play.
140  *
141  * RETURNSthe character value of the player that won or a value that
142  *   indicates a tie.
143  ********************************************************************************/
144 char CheckWin(const char boardAr[][3]);// IN -tic tac toe board
145                                        // Done
146
147
148 /*******************************************************************************
149  * OutputWinner
150  *   This function receives as input a character indicating which player won
151  *   or if the game was a tie and outputs an appropriate message. This function
152  *   does not return anything as it simply outputs the appropriate message to
153  *   the screen.
154  *
155  * RETURNS: nothing
156  * Displays the winner's name
157  ********************************************************************************/
158 void OutputWinner(char &wonPlayer,  // IN  -represents the winner or a value
159                                     //  indicating a tied game.
160                   string playerX,   //OUT -player X's name
161                   string playerO,   //OUT -player O'x name
162                   char   tokenChoice,
163                   char   compToken,
164                   int    option);
165
```

```
166
167
168 #endif /* MYHEADER_H_ */
169
```