Assignment 3: Gomoku

Due: 20:00, Wed 9 Nov 2016 Full marks: 100

1 Introduction

In this assignment, you will use two-dimensional array to implement a game called *Gomoku* (五子棋). It is played on a $n \times n$ game board. (We assume $n = 13$ in this assignment.) Two players O and X take turns to put their game discs into one unoccupied square of the board. The player who first forms a line of five or more consecutive discs horizontally —, vertically ｜, or diagonally ╲╱ wins the game. ("Go" in Gomoku means "five" in Japanese.) The game is a draw when the board is full but no player wins. Figure 1 shows an example configuration of Gomoku. The character '.' denotes an empty square. The rows and columns are named in numbers (0–12) and letters (A–M) respectively. If Player X puts a disc to the square G7, then a diagonal line ╱ of five consecutive X's will be formed and thus X will win.

```
   A B C D E F G H I J K L M
 0 . . . . . . . . . . . . .
 1 . . . . . . . . . . . . .
 2 . . . . . . . . . . . . .
 3 . . . . . . . . . . . . .
 4 . . . . . . . . . . . . .
 5 . . . . . . . X X . . . .
 6 . . . . . . . X O . . . .
 7 . . . . . O . O O . . . .
 8 . . . . . X X O O . . . .
 9 . . . . X O O X X . . . .
10 . . . . . . . . . . . . .
11 . . . . . . . . . . . . .
12 . . . . . . . . . . . . .
```

Figure 1: An Example Game Configuration of Gomoku

2 Program Specification

2.1 Game Board

You can use a two-dimensional array of `char` to represent the game board.

```
const int N = 13;
…
char board[N][N];
```

The array elements `board[0][0]`, `board[0][12]`, `board[12][0]`, and `board[12][12]` denote the four corner squares A0, M0, A12, and M12 respectively.

2.2 Game Flow

1. The game starts with an empty game board. Player O takes the first turn.

2. In each move, you should prompt the player to enter two inputs denoting the square location to be placed. You can assume that *the first input is always a character* and *the second input is always an integer*.
   - A user input is invalid if: (a) it is not a proper cell location (i.e., rows 0–12 and columns A–M, *capital letters only*), *or* (b) the input location is already occupied.
   - You should warn the player about invalid inputs and prompt the same player to enter again until a valid input is entered.
3. Update the board by putting a disc to the appropriate square.
4. Determine if the current player has connected five or above.
5. Repeat Steps 2–4 until the game is finished. Alternate Players O and X in each round.
6. Once the game is finished, display the message "Player O wins!", "Player X wins!", or "Draw game!" accordingly.

2.3 Other Notes

- You are *not allowed to use global variables* in your program. (That is, you cannot declare any variables outside any functions.) Nonetheless, `const` ones (e.g., N) do not count.
- Your program should be decomposed into *at least four functions* (including `main()`). At least *two functions* should have *array parameter(s)*.

3 Program Output

The following shows some sample output of the program. The **bold blue** text is user input and the other text is the program output. You can try the provided sample program for other input. *Your program output should be exactly the same as the sample program* (i.e., same text, same symbols, same letter case, same number of spaces, etc.). Otherwise, it will be considered as *wrong*, even if you have computed the correct result.

```
   A B C D E F G H I J K L M
 0 . . . . . . . . . . . . .
 1 . . . . . . . . . . . . .
 2 . . . . . . . . . . . . .
 3 . . . . . . . . . . . . .
 4 . . . . . . . . . . . . .
 5 . . . . . . . . . . . . .
 6 . . . . . . . . . . . . .
 7 . . . . . . . . . . . . .
 8 . . . . . . . . . . . . .
 9 . . . . . . . . . . . . .
10 . . . . . . . . . . . . .
11 . . . . . . . . . . . . .
12 . . . . . . . . . . . . .
Player O, make a move: G -1↵
Invalid input. Try again!
Player O, make a move: P 8↵
Invalid input. Try again!
Player O, make a move: E 6↵
```

```
   A B C D E F G H I J K L M
 0 . . . . . . . . . . . . .
 1 . . . . . . . . . . . . .
 2 . . . . . . . . . . . . .
 3 . . . . . . . . . . . . .
 4 . . . . . . . . . . . . .
 5 . . . . . . . . . . . . .
 6 . . . . O . . . . . . . .
 7 . . . . . . . . . . . . .
 8 . . . . . . . . . . . . .
 9 . . . . . . . . . . . . .
10 . . . . . . . . . . . . .
11 . . . . . . . . . . . . .
12 . . . . . . . . . . . . .
Player X, make a move: C 11↵
   A B C D E F G H I J K L M
 0 . . . . . . . . . . . . .
 1 . . . . . . . . . . . . .
 2 . . . . . . . . . . . . .
 3 . . . . . . . . . . . . .
 4 . . . . . . . . . . . . .
 5 . . . . . . . . . . . . .
 6 . . . . O . . . . . . . .
 7 . . . . . . . . . . . . .
 8 . . . . . . . . . . . . .
 9 . . . . . . . . . . . . .
10 . . . . . . . . . . . . .
11 . . X . . . . . . . . . .
12 . . . . . . . . . . . . .
Player O, make a move: C 11↵
Invalid input. Try again!
Player O, make a move: F 5↵
   A B C D E F G H I J K L M
 0 . . . . . . . . . . . . .
 1 . . . . . . . . . . . . .
 2 . . . . . . . . . . . . .
 3 . . . . . . . . . . . . .
 4 . . . . . . . . . . . . .
 5 . . . . . O . . . . . . .
 6 . . . . O . . . . . . . .
 7 . . . . . . . . . . . . .
 8 . . . . . . . . . . . . .
 9 . . . . . . . . . . . . .
10 . . . . . . . . . . . . .
11 . . X . . . . . . . . . .
12 . . . . . . . . . . . . .
```

```
    .
    .     } (Many moves are skipped to save space. See Blackboard for full version.)
    .
   A B C D E F G H I J K L M
 0 . . . . . . . . . . . . .
 1 . . . . . . . . . . . . .
 2 . X . . . . . . . . . . .
 3 X . . . . . . O . . . . X
 4 . . . . . . O . . . . . .
 5 . . . . . O . . . . . . .
 6 . . . . O . . . . . . . .
 7 . . . . O . . . . . . . .
 8 . . . . . . . . . . . . .
 9 . . . . . . . . . . . . .
10 . . . . . . . . . . . . .
11 . . X . . . . . . . . . .
12 . . . . . . . X . . . . .
Player O, make a move: D 7↵
   A B C D E F G H I J K L M
 0 . . . . . . . . . . . . .
 1 . . . . . . . . . . . . .
 2 . X . . . . . . . . . . .
 3 X . . . . . . O . . . . X
 4 . . . . . . O . . . . . .
 5 . . . . . O . . . . . . .
 6 . . . . O . . . . . . . .
 7 . . . O O . . . . . . . .
 8 . . . . . . . . . . . . .
 9 . . . . . . . . . . . . .
10 . . . . . . . . . . . . .
11 . . X . . . . . . . . . .
12 . . . . . . . X . . . . .
Player O wins!
```

4 Submission and Marking

- Your program file name should be gomoku.cpp. Submit the file in Blackboard. (https://elearn.cuhk.edu.hk/).
- Insert your name, student ID, and e-mail address as comments at the top of your source file.
- Besides the above information, your program should further include suitable comments as documentation.
- You can submit your assignment multiple times. Only the latest submission counts.
- Your program should be *free of compilation errors and warnings*.
- *Plagiarism is strictly monitored and heavily punished if proven.* Lending your work to others is subjected to the same penalty as the copier.