**10902 CPP Midterm Exam**

|  |
| --- |
| **Contributor︰Yen-Chen Chiu** |
| **Subject：Connect Four** |
| **Main testing concept：**   |  |  | | --- | --- | | **Basics** | **Functions** | | ■ C++ BASICS  ■ FLOW OF CONTROL  ■ FUNCTION BASICS  □ PARAMETERS AND OVERLOADING  ■ ARRAYS  □ STRUCTURES AND CLASSES  □ CONSTRUCTORS AND OTHER TOOLS  □ OPERATOR OVERLOADING, FRIENDS, AND REFERENCES  □ STRINGS  □ POINTERS AND DYNAMIC ARRAYS | □ SEPARATE COMPILATION AND NAMESPACES  □ STREAMS AND FILE I/O  □ RECURSION  □ INHERITANCE  □ POLYMORPHISM AND VIRTUAL FUNCTIONS  □ TEMPLATES  □ LINKED DATA STRUCTURES  □ EXCEPTION HANDLING  □ STANDARD TEMPLATE LIBRARY  □ PATTERNS AND UML | |
| **Description：**  Connect Four is a tabletop game in which two players take alternating turns placing a checker into a vertically suspended grid. The first player to connect four of their checkers vertically, or horizontally, or diagonally wins.  In this test, you need to implement this game, that the players can place checkers into the line (column) by input the line numbers, and for each checker placed, determine if the game is over. When game’s over, all the checkers are removed from the grid, and start a new game.  The game has a 7x6 grid and checkers of two colors, YELLOW checkers and RED checkers. YELLOW takes the first turn, then RED, then YELLOW… and so on, until the game is over. Each turn the checker is placed, then falls down on the top of the stack of other checkers in the same column, if the 4 checkers connected vertically, horizontally, or diagonally, then the player of that turn wins, and the game is over.  The game might be drawn when all the grid is filled, but no player wins.  The player might choose a column which is already filled, or number is not in 1 to 7, you need to notify the player to choose again.  **Input：**  The game start with an empty grid. And for each turn, the player input the line number chosen from 1 to 7, imply the position where the checker should be placed.  **Output：**   * At the end of each turn, check if the game is over,  prints “YELLOW wins!” or “RED wins!”, or “Draw!”. Then start next game’s first turn. * If the player chooses a column which is already full, prints “The line is full, choose again:”. * If the player inputs a number which is not in 1 to 7, prints “The line isn't exist, choose again:”.   **Sample Input / Output :**   |  |  | | --- | --- | | **Sample Input** | **Sample Output** | | 1  2  1  2  1  2  1  1  2  3  2  4  2  5  2 | **(input numbers are for better visualization, you don’t need to print those number.)**  1  2  1  2  1  2  1  YELLOW wins!  1  2  3  2  4  2  5  2  RED wins! | | 1  1  1  1  1  1  1  2  2  3  3  4 | 1  1  1  1  1  1  1  The line is full, choose again:  2  2  3  3  4  YELLOW wins! | | -1  0  10  4  4  4  4  4  4  3  2  3  5  5  3  5  2  5  5 | -1  The line isn't exist, choose again:  0  The line isn't exist, choose again:  10  The line isn't exist, choose again:  4  4  4  4  4  4  3  2  3  5  5  3  5  2  5  5  RED wins! | |
| **■** **Easy, only basic programming syntax and structure are required.**  **□ Medium, multiple programming grammars and structures are required.**  **□ Hard, need to use multiple program structures or complex data types.** |
| **Expected solving time:**  20 minutes |
| **Other notes:** |