OOP (Object Oriented Programming) Lab

LAB REPORT # 2

Semester: 2nd Semester

Section: C

Submitted To:

Mr. Muhammad Husnain

Submitted By:

Name: Muhammad Afzal

Roll No: 22-CS-035

Task 1:

```cpp
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

int main()
{
    const int SIZE = 100;
    int arr[SIZE];

    

    srand(time(nullptr));


    for (int i = 0; i < SIZE; i++)
        arr[i] = rand() % 1000 + 1;


    int minVal = arr[0];
    int maxVal = arr[0];


    for (int i = 1; i < SIZE; i++)
    {
        if (arr[i] < minVal)
            minVal = arr[i];
        if (arr[i] > maxVal)
            maxVal = arr[i];
    }


    cout << "Minimum value: " << minVal << endl;
    cout << "Maximum value: " << maxVal << endl;

    return 0;
}

```

Output:

```
Minimum value: 27
Maximum value: 973
```

Task 2:

```cpp
#include <iostream>
#include <cstring>

using namespace std;

const int ROWS = 3;
const int COLS = 3;

void displayBoard(char board[][COLS]) {
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            cout << board[i][j] << " ";
        }
        cout << endl;
    }
}

bool isWin(char board[][COLS], char player) {
    for (int i = 0; i < ROWS; i++) {
        if (board[i][0] == player && board[i][1] == player && board[i][2] == player) {
            return true;
        }
    }

    for (int j = 0; j < COLS; j++) {
        if (board[0][j] == player && board[1][j] == player && board[2][j] == player) {
            return true;
        }
    }

    if (board[0][0] == player && board[1][1] == player && board[2][2] == player) {
        return true;
    }

    if (board[0][2] == player && board[1][1] == player && board[2][0] == player) {
        return true;
    }

    return false;
}

bool isTie(char board[][COLS]) {
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            if (board[i][j] == '-') {
                return false;
            }
        }
    }
    return true;
}

void playGame() {
    char board[ROWS][COLS];
    memset(board, '-', sizeof(board)); // Initialize board to all '-'

    char currentPlayer = 'X';
    bool isGameDone = false;

    while (!isGameDone) {
        displayBoard(board);

        int row, col;
        cout << "Player " << currentPlayer << " Enter your move (row col): ";
        cin >> row >> col;

        if (row < 0 || row >= ROWS || col < 0 || col >= COLS) {
            cout << "Invalid move, try again." << endl;
            continue;
        }

        if (board[row][col] != '-') {
            cout << "That cell is already occupied, try again." << endl;
            continue;
        }
```

```
78
79          if (isWin(board, currentPlayer)) {
80              cout << "Player " << currentPlayer << " has won the game!" << endl;
81              isGameDone = true;
82          }
83          else if (isTie(board)) {
84              cout << "The game is a tie!" << endl;
85              isGameDone = true;
86          }
87          else {
88              // Switch to the other player
89              currentPlayer = (currentPlayer == 'X') ? 'O' : 'X';
90          }
91      }
92
93      displayBoard(board);
94  }
95
96  int main() {
97      cout <<"Welcome to Tic Tac Toe!" << endl;
98
99      while (true) {
100         playGame();
101
102         char playAgain;
103         cout << "Play again? (y/n): ";
104         cin >> playAgain;
105
106         if (playAgain != 'y' && playAgain != 'Y') {
107             break;
108         }
109     }
110
111     cout << "Thanks for playing!" << endl;
112
113     return 0;
114 }
115
```

Output:

```
- X -
- - -
- - -
Player O Enter your move (row col):
```

```
Player O has won the game!
O X X
- O -
- X O
Play again? (y/n):
```