Task 1: Number Guessing Game

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

using namespace std;

void playNumberGuessingGame() {
    srand(time(0));
    int numberToGuess = rand() % 100 + 1;
    int guess;
    cout << "Guess a number between 1 and 100: ";
    while (true) {
        cin >> guess;
        if (guess > numberToGuess) {
            cout << "Too high! Try again: ";
        } else if (guess < numberToGuess) {
            cout << "Too low! Try again: ";
        } else {
            cout << "Congratulations! You guessed the number!" << endl;
            break;
        }
    }
}

int main() {
```

```
    playNumberGuessingGame();
    return 0;
}
```

Task 2: Simple Calculator

```
#include <iostream>

using namespace std;

void simpleCalculator() {
    double num1, num2, result;
    char operation;
    cout << "Enter first number: ";
    cin >> num1;
    cout << "Enter an operator (+, -, *, /): ";
    cin >> operation;
    cout << "Enter second number: ";
    cin >> num2;

    switch (operation) {
        case '+':
            result = num1 + num2;
            break;
        case '-':
            result = num1 - num2;
            break;
```

```
        case '*':
            result = num1 * num2;
            break;
        case '/':
            if (num2 != 0) {
                result = num1 / num2;
            } else {
                cout << "Error! Division by zero." << endl;
                return;
            }
            break;
        default:
            cout << "Invalid operator!" << endl;
            return;
    }
    cout << "Result: " << result << endl;
}

int main() {
    simpleCalculator();
    return 0;
}
```

Task 3: Tic-Tac-Toe Game

```
#include <iostream>
#include <vector>
```

```
using namespace std;

void displayBoard(const vector<char>& board) {
    cout << "\n";
    for (int i = 0; i < 9; i++) {
        cout << board[i] << " ";
        if ((i + 1) % 3 == 0) cout << "\n";
    }
}

bool checkWin(const vector<char>& board, char player) {
    const int winPatterns[8][3] = {
        {0, 1, 2}, {3, 4, 5}, {6, 7, 8}, // Rows
        {0, 3, 6}, {1, 4, 7}, {2, 5, 8}, // Columns
        {0, 4, 8}, {2, 4, 6}             // Diagonals
    };

    for (const auto& pattern : winPatterns) {
                if (board[pattern[0]] == player && board[pattern[1]] == player &&
board[pattern[2]] == player) {
            return true;
        }
    }
    return false;
}
```

```
bool checkDraw(const vector<char>& board) {
    for (char cell : board) {
        if (cell != 'X' && cell != 'O') return false;
    }
    return true;
}

void ticTacToeGame() {
    vector<char> board(9);
    for (int i = 0; i < 9; ++i) board[i] = '1' + i;

    char currentPlayer = 'X';
    while (true) {
        displayBoard(board);
        int move;
        cout << "Player " << currentPlayer << ", enter your move (1-9): ";
        cin >> move;
        if (move < 1 || move > 9 || board[move - 1] == 'X' || board[move - 1] == 'O') {
            cout << "Invalid move. Try again.\n";
            continue;
        }
        board[move - 1] = currentPlayer;
        if (checkWin(board, currentPlayer)) {
            displayBoard(board);
            cout << "Player " << currentPlayer << " wins!\n";
            break;
        }
```

```
        if (checkDraw(board)) {
            displayBoard(board);
            cout << "It's a draw!\n";
            break;
        }
        currentPlayer = (currentPlayer == 'X') ? 'O' : 'X';
    }
}

int main() {
    ticTacToeGame();
    return 0;
}
```