WORK SHEET 1

Manish Mahato

The British College
Student ID: 23085136

Worksheet 1

Task 1

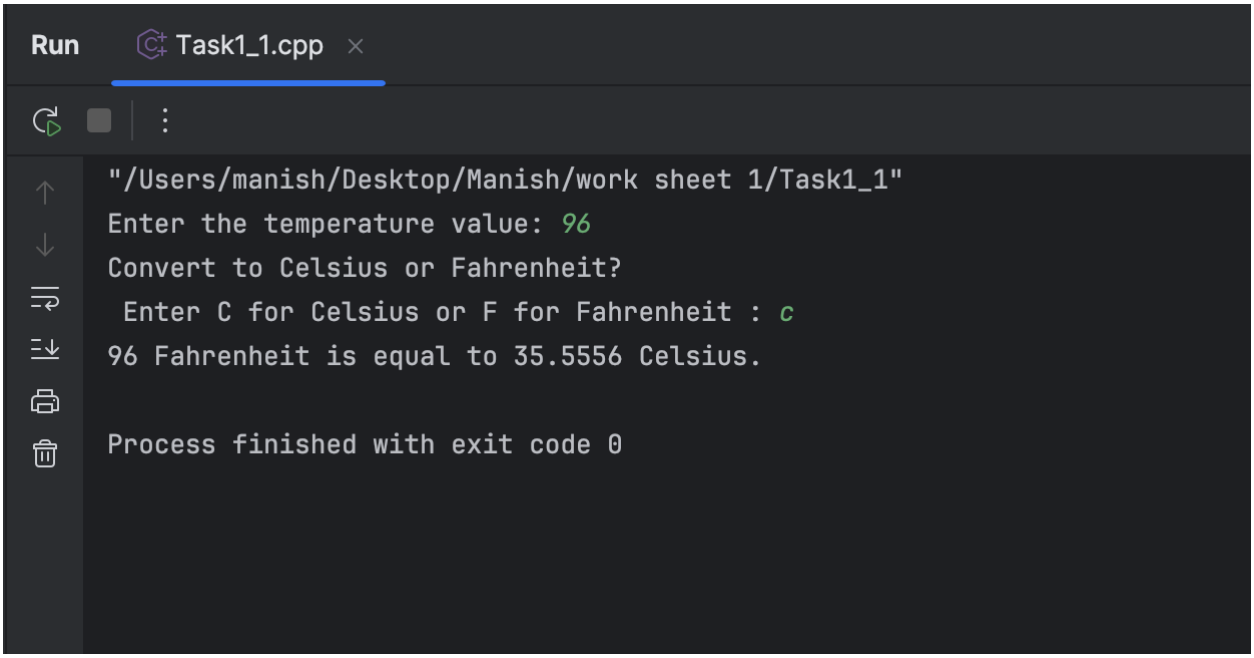
1. Write a program that takes a temperature value from the user. It should then allow the user to choose between Celsius (C) and Fahrenheit (F) for conversion. After the user selection, it should then convert the entered temperature to the chosen scale and display the result. Use appropriate data types for temperature and handle error like non-numeric input.
   Use the following formula for conversion:
   F = (C x 9/5) + 32
   C = (F - 32) x 5/9

Source code for this:

```
#include <iostream>
using namespace std;

void convertTemperature() {
    double temp, converted_temp;
    char choice;

    cout << "Enter the temperature value: ";
    if (!(cin >> temp)) {
        cout << "Invalid input. Please enter a numeric value for
temperature." << endl;
        return;
    }

    cout << "Convert to Celsius or Fahrenheit? \n Enter C for Celsius or F
for Fahrenheit : ";
    cin >> choice;

    choice = toupper(choice);

    if (choice == 'C') {
        converted_temp = (temp - 32) * 5 / 9;
        cout << temp << " Fahrenheit is equal to " << converted_temp << "
Celsius." << endl;
    } else if (choice == 'F') {
        converted_temp = (temp * 9 / 5) + 32;
        cout << temp << " Celsius is equal to " << converted_temp << "
Fahrenheit." << endl;
    } else {
        cout << "Invalid choice. Please enter 'C' or 'F'." << endl;
    }
}

int main() {
```

```
    convertTemperature();
    return 0;
}
```

Output:



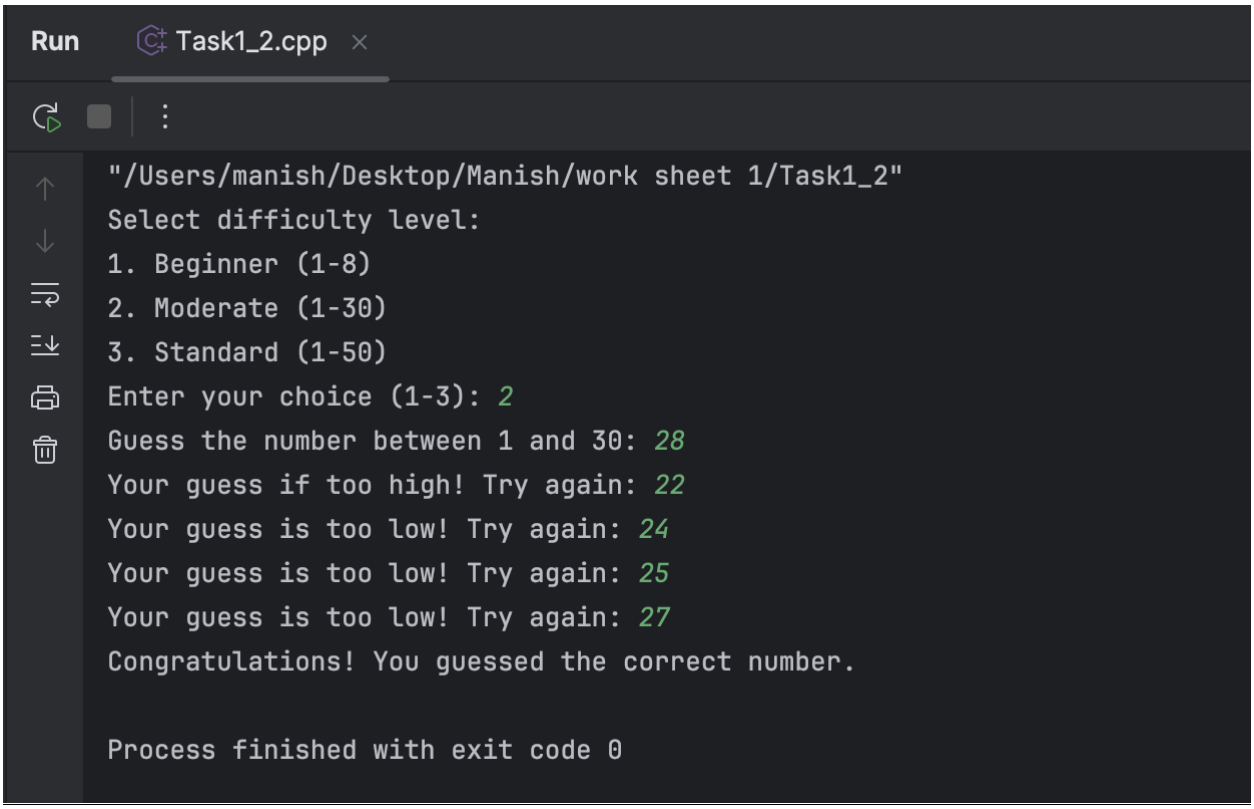
2. Write a C++ program to implement a number guessing game with different difficulty levels. Easy difficulty ranges from 1-8, medium from 1-30, hard from 1-50.Then,generate a random number to check if the guess is correct based on the user's selection.

Source code for this:

```
#include <iostream>

using namespace std;

class guessgame {
private:
    int randomnumber;
    int maximumnumber;
public:

    void numberGuessingGame (){
        int difficulty, range, number, guess;

        cout << "Select difficulty level:\n";
        cout << "1. Beginner (1-8)\n";
        cout << "2. Moderate (1-30)\n";
        cout << "3. Standard (1-50)\n";
```

```
        cout << "Enter your choice (1-3): ";
        cin >> difficulty;

        if (difficulty == 1) {
            range = 8;
        } else if (difficulty == 2) {
            range = 30;
        } else if (difficulty == 3) {
            range = 50;
        } else {
            cout << "Invalid selection. Please choose between the range." <<
endl;
            return;
        }

        srand(time(0));
        number = rand() % range + 1;

        cout << "Guess the number between 1 and " << range << ": ";
        while (true) {
            cin >> guess;
            if (guess < 1 || guess > range) {
                cout << "Out of range! Please guess between 1 and " << range
<< "." << endl;
                continue;
            }
            if (guess == number) {
                cout << "Congratulations! You guessed the correct number." <<
endl;
                break;
            } else if (guess < number) {
                cout << "Your guess is too low! Try again: ";
            } else {
                cout << "Your guess if too high! Try again: ";
            }
        }
    }
};
int main() {
    guessgame Game;
    Game.numberGuessingGame();
    return 0;
}
```

Output:



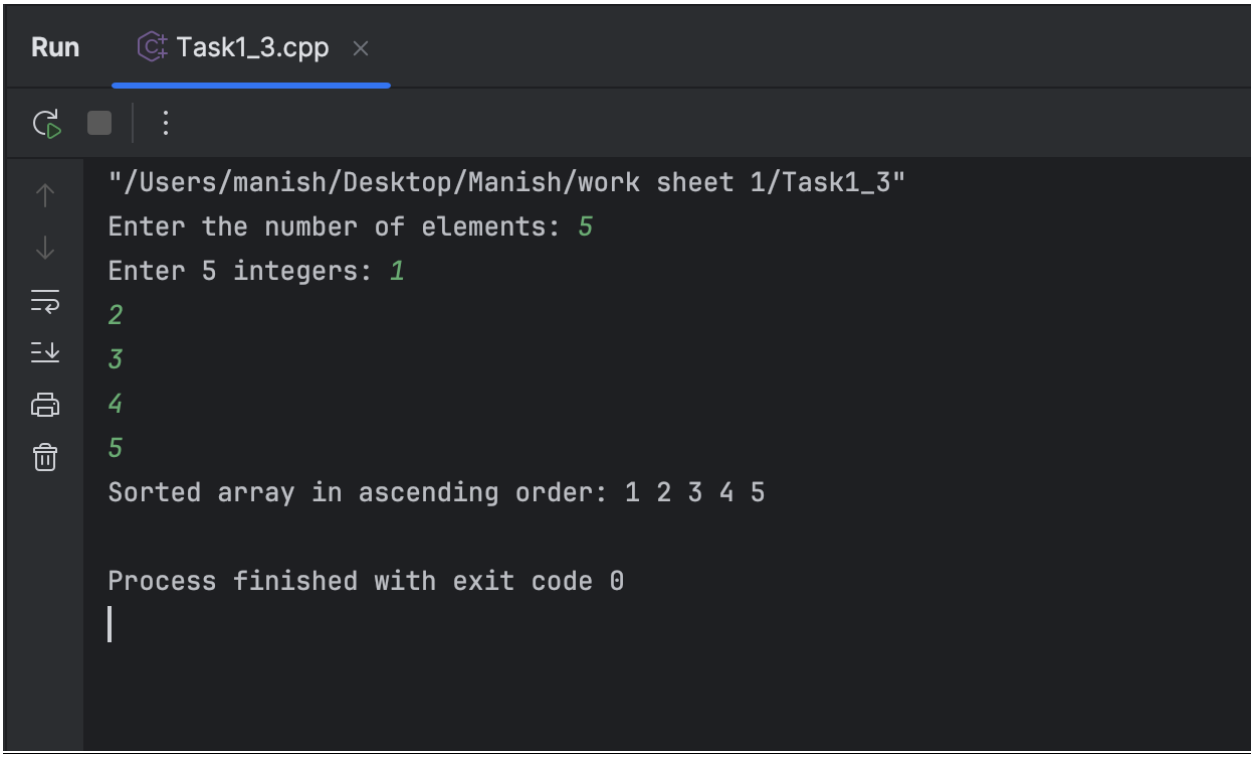
3. Write a program that reads an array of integer numbers from the user and sorts the numbers in the ascending order.

Source code for this:

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

class ArraySorter {
private:
    vector<int> arr;

public:
    void getInput() {
        int n;
        cout << "Enter the number of elements: ";
        cin >> n;

        if (n <= 0) {
            cout << "Invalid input. The number of elements must be greater
than 0." << endl;
            exit(0);
        }

        arr.resize(n);
```

```
        cout << "Enter " << n << " integers: ";
        for (int i = 0; i < n; i++) {
            cin >> arr[i];
        }
    }

    void sortArray() {
        sort(arr.begin(), arr.end());
    }

    void displayArray() {
        cout << "Sorted array in ascending order: ";
        for (int num : arr) {
            cout << num << " ";
        }
        cout << endl;
    }

    void process() {
        getInput();
        sortArray();
        displayArray();
    }
};

int main() {
    ArraySorter sorter;
    sorter.process();
    return 0;
}
```

Output:



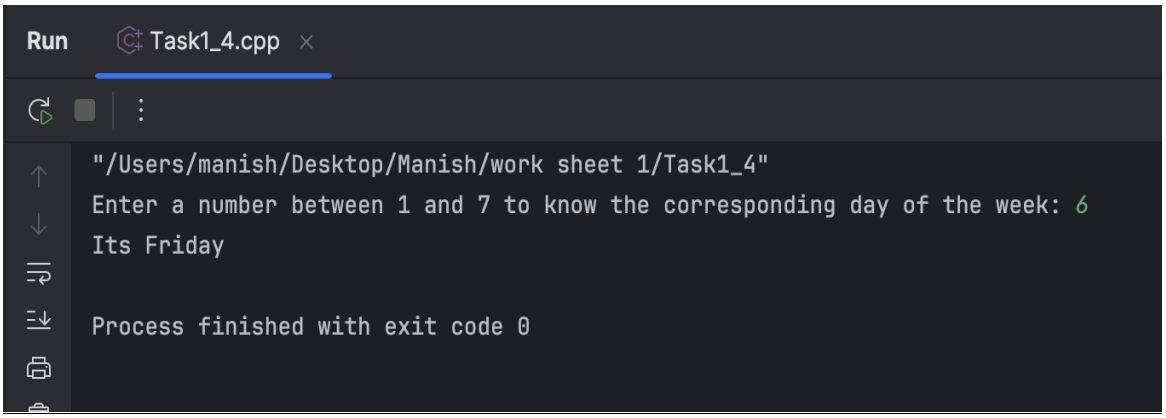
4. Write a program that reads a number from the user and based on the user input, it says what day of the week it is, Sundays being 1 and Saturdays being 7. You system should give appropriate response for invalid input entries.

Source code for this:

```
#include <iostream>
using namespace std;

class DayOfWeek
{
public:
    void getWeekDay(int day)
    {
        switch (day)
        {
            case 1:
                cout << "Its Sunday" << endl;
            break;
            case 2:
                cout << "Its Monday" << endl;
            break;
            case 3:
                cout << "Its Tuesday" << endl;
            break;
            case 4:
                cout << "Its Wednesday" << endl;
```

```
            break;
            case 5:
                cout << "Its Thursday" << endl;
            break;
            case 6:
                cout << "Its Friday" << endl;
            break;
            case 7:
                cout << "Its Saturday" << endl;
            break;
            default:
                cout << "Invalid input. Please select between 1 and 7." <<
endl;
            break;
        }
    }
};

void getInput() {
    int day;
    cout << "Enter a number between 1 and 7 to know the corresponding day of
the week: ";
    cin >> day;

    DayOfWeek d;
    d.getWeekDay(day);
}

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

Output:



Task 2:

1. Create a program that takes a positive integer as input and determines whether it's a "bouncy number". A bouncy number is one where the digits neither consistently increase nor consistently decrease when read from left to right. For example:

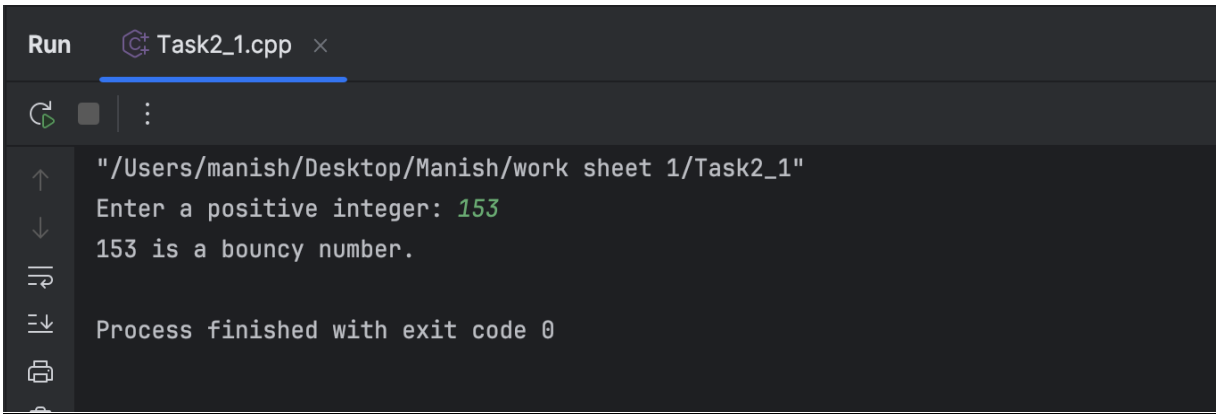
- 123 is NOT bouncy (digits consistently increase)
- 321 is NOT bouncy (digits consistently decrease)
- 120 is bouncy (neither consistently increasing nor decreasing)

Source code for this:

```
#include <iostream>
#include <vector>
using namespace std;

class BouncyNumber
{
public:
    bool isBouncy(int number)

    {
        vector<int> digits = getDigits(number);

        if (digits.size() < 3)
            {
            return false;
        }

        bool increasing = false;
        bool decreasing = false;

        for (size_t i = 1; i < digits.size(); ++i)
            {
            if (digits[i] > digits[i - 1])
            {
                increasing = true;
            } else if (digits[i] < digits[i - 1])
            {
                decreasing = true;
            }

            if (increasing && decreasing)
                {
                return true;
            }
        }


        return false;
    }

private:
```

```
    vector<int> getDigits(int number)
    {
        vector<int> digits;
        while (number > 0)
        {
            digits.insert(digits.begin(), number % 10);
            number /= 10;
        }
        return digits;
    }
};

int main()
{
    int number;
    cout << "Enter a positive integer: ";
    cin >> number;

    BouncyNumber bouncyChecker;

    if (bouncyChecker.isBouncy(number))
        {
        cout << number << " is a bouncy number." << endl;
        }
    else
        {
        cout << number << " is not a bouncy number." << endl;
        }

    return 0;
}
```

Output:



Task 3

1. Write a program that manages a cinema ticket booking system. The program should display a 5x5 seating arrangement where:
    1. Available seats are marked with 'O'
    2. Booked seats are marked with 'X'

Program should:

1. Display the current seating arrangement
2. Ask user for row and column number (1-5) for booking
3. Mark that seat as booked ('X')
4. Show updated seating after each booking
5. Display error if user selects already booked seat
6. Display error if user enters invalid row/column numbers

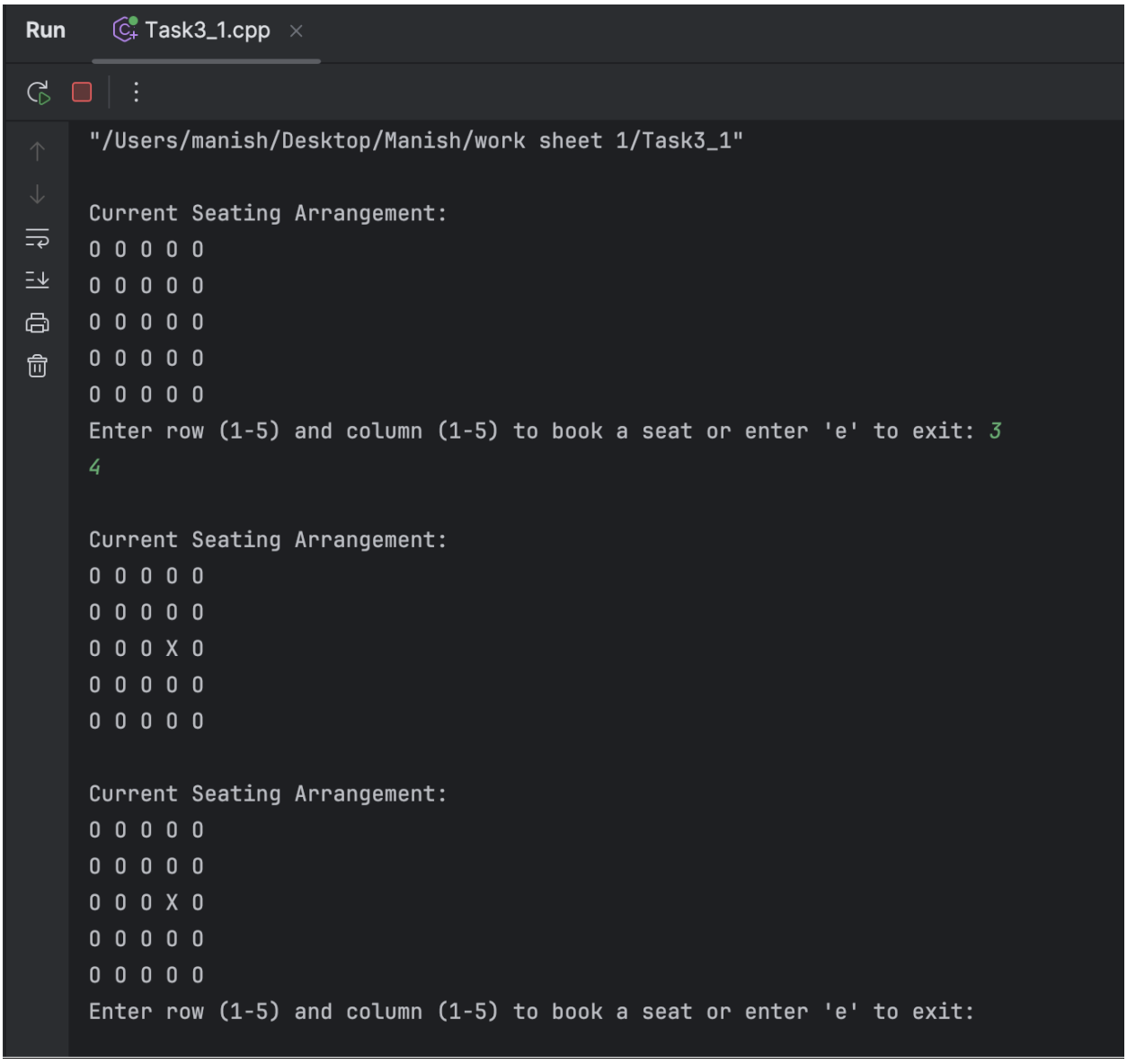
Source code for this:

```
#include <iostream>
#include <limits>
using namespace std;


class Cinema
{
private:
    char seats[5][5];

public:

    Cinema()
    {
        for (int i = 0; i < 5; ++i)
            {
            for (int j = 0; j < 5; ++j)
            {
                seats[i][j] = 'O';
            }
        }
    }

    void displaySeating()
    {
        cout << "\nCurrent Seating Arrangement:\n";
```

```
        for (int i = 0; i < 5; ++i)
            {
            for (int j = 0; j < 5; ++j)
            {
                cout << seats[i][j] << " ";
            }
            cout << endl;
        }
    }

    bool bookSeat(int row, int col)
    {
        row -= 1;
        col -= 1;

        if (row < 0 || row >= 5 || col < 0 || col >= 5)
            {
            cout << "Error: Invalid seat number. Please enter row and column
between 1 and 5.\n";
            return false;
        }

        if (seats[row][col] == 'X')
            {
            cout << "Error: This seat is already booked. Please choose a
different seat.\n";
            return false;
        }

        seats[row][col] = 'X';
        return true;
    }
};


void Bookingsystem(Cinema &cinema)
{
    int row, col;
    bool bookingSuccessful;
    char exitOption;


    while (true)
        {
        cinema.displaySeating();

        cout << "Enter row (1-5) and column (1-5) to book a seat or enter 'e'
to exit: ";


        if (!(cin >> row))
            {

            cin.clear();
            cin >> exitOption;
            if (exitOption == 'e')
            {
```

```
                cout << "Exiting the system.\n";
                break;
            }

            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            continue;
        }


        if (!(cin >> col))
            {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "Error: Invalid input for column. Please enter valid
numbers.\n";
            continue;
        }


        bookingSuccessful = cinema.bookSeat(row, col);


        if (bookingSuccessful)
            {
            cinema.displaySeating();
        }
    }
}

int main()
{

    Cinema cinema;


    Bookingsystem(cinema);

    return 0;
}
```

Output:


Run
Task3_1.cpp
"/Users/manish/Desktop/Manish/work sheet 1/Task3_1"
Current Seating Arrangement:
O O O O O
O O O O O
O O O O O
O O O O O
O O O O O
Enter row (1-5) and column (1-5) to book a seat or enter 'e' to exit: 3
4
Current Seating Arrangement:
O O O O O
O O O O O
O O O X O
O O O O O
O O O O O
Current Seating Arrangement:
O O O O O
O O O O O
O O O X O
O O O O O
O O O O O
Enter row (1-5) and column (1-5) to book a seat or enter 'e' to exit: