The British College

WORKSHEET- 1

Submitted By: Sandhya Aryal

Student ID: 24000860

Cyber Security and Digital Forensics

Github Link:

https://github.com/Sandhyaaaa1/Cpp_Worksheet

Task 1 : Programming Exercises:[Data types and Conditional Statements]

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:

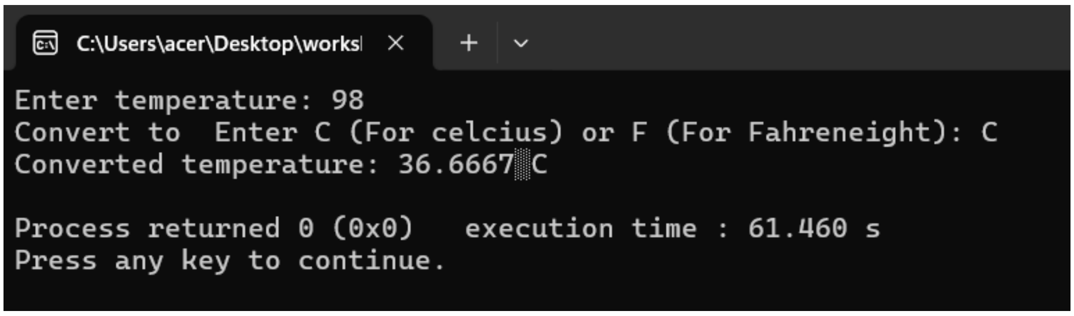
F = (C x 9/5) + 32

C = (F - 32) x 5/9 [10 marks]

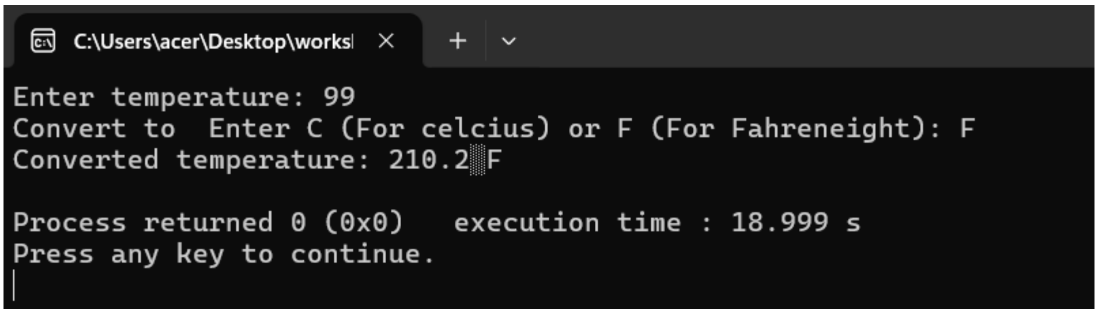
```
#include <iostream>

#include <limits>

using namespace std;

class ConvertTemperature

{

public:

  static double toFahrenheit(double temp)

  {
      return (temp * 9.0 / 5.0) + 32;

  }

  static double toCelsius(double temp)

  {

      return (temp - 32) * 5.0 / 9.0;

  }
};

int main()

{
  double temp;

  char choice;

  cout << "Enter temperature: ";

  while (!(cin >> temp))

      {

          cin.clear();

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

          cout << "Your input is Invalid. Enter a numeric value: ";
```

```
      }

    cout << "Convert to  Enter C (For celcius) or F (For Fahreneight): ";

    cin >> choice;

    choice = toupper(choice);



    if (choice == 'C')



      cout << "Converted temperature: " << ConvertTemperature::toCelsius(temp) << "°C" << endl;

    else if (choice == 'F')

      cout << "Converted temperature: " << ConvertTemperature::toFahrenheit(temp) << "°F" << endl;

    else

      cout << "Your choice is Invalid." << endl;

    return 0;
}
```

OUTPUT:



```
Enter temperature: 98
Convert to  Enter C (For celcius) or F (For Fahreneight): C
Converted temperature: 36.6667░C

Process returned 0 (0x0)   execution time : 61.460 s
Press any key to continue.
```



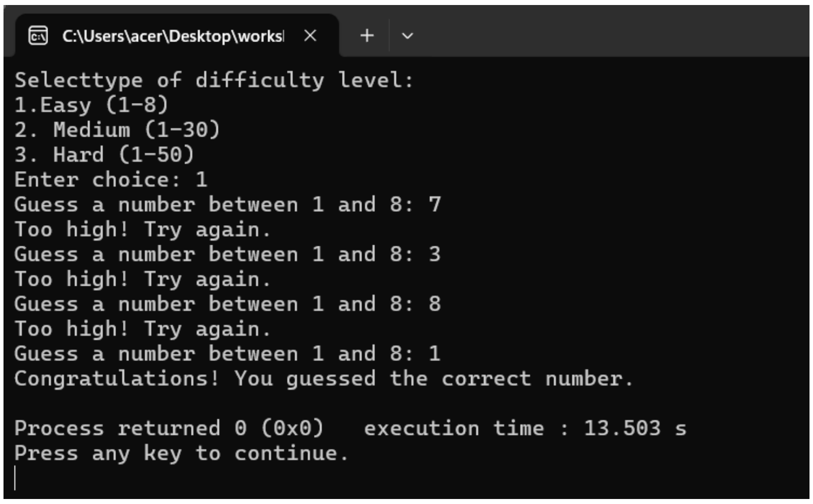
```
Enter temperature: 99
Convert to  Enter C (For celcius) or F (For Fahreneight): F
Converted temperature: 210.2░F

Process returned 0 (0x0)   execution time : 18.999 s
Press any key to continue.
```

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. [10 marks]

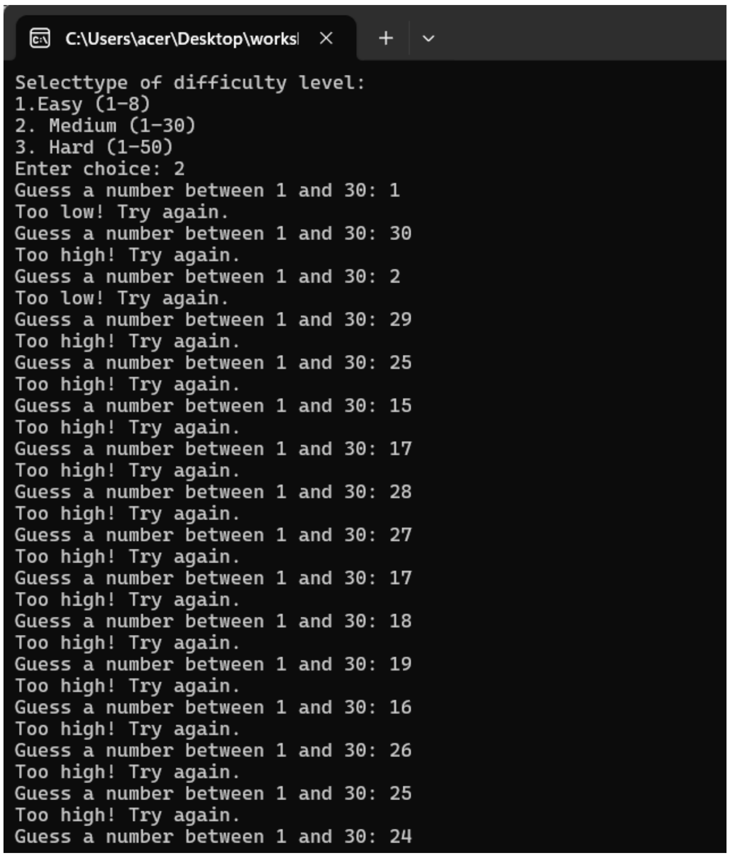
```
#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

class NumberGenerator
{

public:

  static int generateRandomNumber(int maxRange)

  {
    srand(time(0));

    return rand() % maxRange + 1;
    }
};

class UserInput

{

public:

    static int getDifficultyLevel()


    {
        int choice;

        cout << "Selecttype of difficulty level:\n1.";
        cout<<"Easy (1-8)\n2.";
        cout<<" Medium (1-30)\n3. Hard (1-50)\nEnter choice: ";

        cin >> choice;

        return choice;
    }

    static int getGuess(int maxRange)



    {
        int guess;

        while (true)

          {

          cout << "Guess a number between 1 and " << maxRange << ": ";


          cin >> guess;

          if (cin.fail() || guess < 1 || guess > maxRange)


             {
                cin.clear();

                cin.ignore(10000, '\n');

                cout << "Try again.Your input is Invalid. " << endl;

             }

           else
             {

             return guess;
          }
    }
  }
};

class NumberGuessingGame

{

private:

  int maxRange;

  int randomNumber;

public:

  NumberGuessingGame(int range) : maxRange(range),
randomNumber(NumberGenerator::generateRandomNumber(range)) {}

  void play()

  {
    while (true)

      {

      int guess = UserInput::getGuess(maxRange);

      if (guess == randomNumber)

      {
        cout << "Congratulations!!! You guessed the correct number." << endl;

        break;
      }

    else if (guess < randomNumber)


      {
        cout << "Try again.It's Too low! " << endl;
      }

    else

      {
        cout << " Try again.It's Too high!" << endl;
      }
    }
  }
};

int main()

{
  int choice = UserInput::getDifficultyLevel();
```

```
    int range;

    switch (choice)

    {
      case 1: range = 8;
      break;

      case 2: range = 30;
      break;

      case 3: range = 50;
      break;

      default:

        cout << "Your choice of number Invalid . Defaulting to Easy (1-8)." << endl;
        range = 8;
    }

    NumberGuessingGame game(range);

    game.play();

    return 0;
  }
```

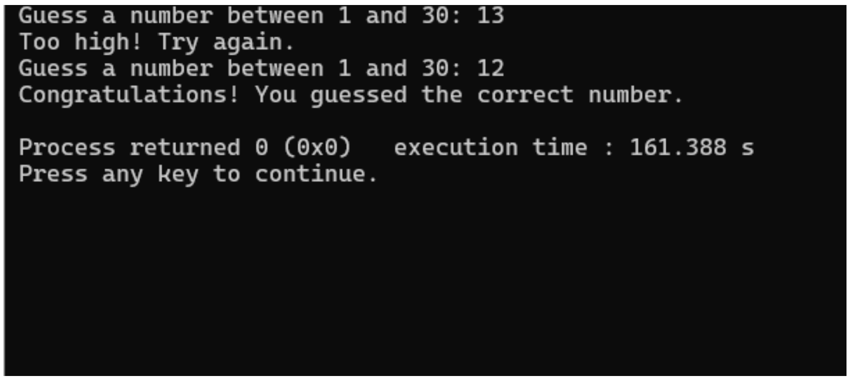
OUTPUT:



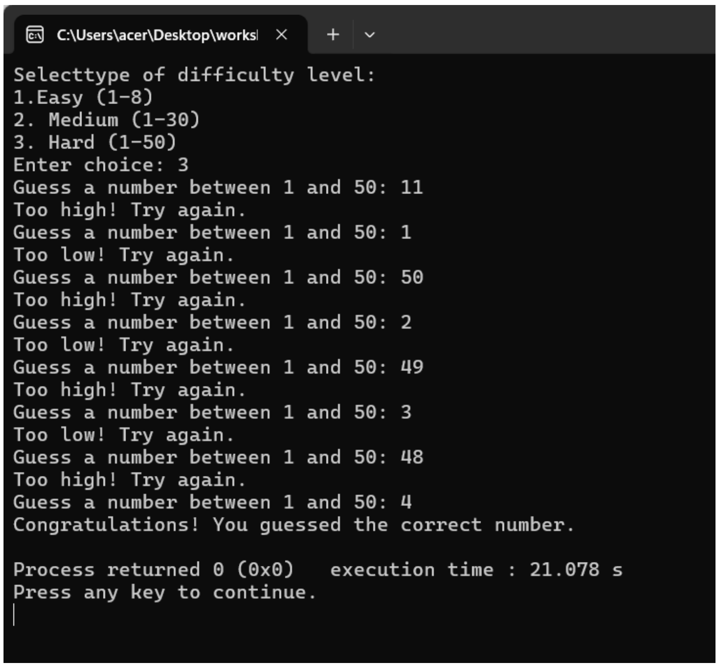
```
Selecttype of difficulty level:
1.Easy (1-8)
2. Medium (1-30)
3. Hard (1-50)
Enter choice: 1
Guess a number between 1 and 8: 7
Too high! Try again.
Guess a number between 1 and 8: 3
Too high! Try again.
Guess a number between 1 and 8: 8
Too high! Try again.
Guess a number between 1 and 8: 1
Congratulations! You guessed the correct number.

Process returned 0 (0x0)   execution time : 13.503 s
Press any key to continue.
```



```
Selecttype of difficulty level:
1.Easy (1-8)
2. Medium (1-30)
3. Hard (1-50)
Enter choice: 2
Guess a number between 1 and 30: 1
Too low! Try again.
Guess a number between 1 and 30: 30
Too high! Try again.
Guess a number between 1 and 30: 2
Too low! Try again.
Guess a number between 1 and 30: 29
Too high! Try again.
Guess a number between 1 and 30: 25
Too high! Try again.
Guess a number between 1 and 30: 15
Too high! Try again.
Guess a number between 1 and 30: 17
Too high! Try again.
Guess a number between 1 and 30: 28
Too high! Try again.
Guess a number between 1 and 30: 27
Too high! Try again.
Guess a number between 1 and 30: 17
Too high! Try again.
Guess a number between 1 and 30: 18
Too high! Try again.
Guess a number between 1 and 30: 19
Too high! Try again.
Guess a number between 1 and 30: 16
Too high! Try again.
Guess a number between 1 and 30: 26
Too high! Try again.
Guess a number between 1 and 30: 25
Too high! Try again.
Guess a number between 1 and 30: 24
```



```
Guess a number between 1 and 30: 13
Too high! Try again.
Guess a number between 1 and 30: 12
Congratulations! You guessed the correct number.

Process returned 0 (0x0)   execution time : 161.388 s
Press any key to continue.
```



```
Selecttype of difficulty level:
1.Easy (1-8)
2. Medium (1-30)
3. Hard (1-50)
Enter choice: 3
Guess a number between 1 and 50: 11
Too high! Try again.
Guess a number between 1 and 50: 1
Too low! Try again.
Guess a number between 1 and 50: 50
Too high! Try again.
Guess a number between 1 and 50: 2
Too low! Try again.
Guess a number between 1 and 50: 49
Too high! Try again.
Guess a number between 1 and 50: 3
Too low! Try again.
Guess a number between 1 and 50: 48
Too high! Try again.
Guess a number between 1 and 50: 4
Congratulations! You guessed the correct number.

Process returned 0 (0x0)   execution time : 21.078 s
Press any key to continue.
```

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

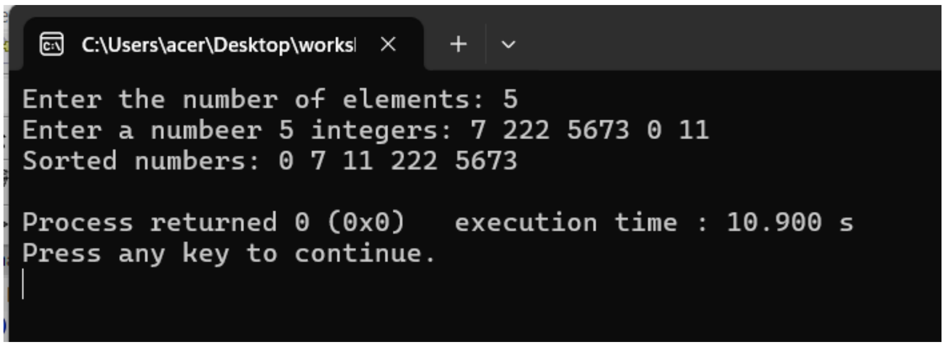
[10 marks]

```
#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

class NumberSorter

{

public:


  static void sortNumbers(vector<int>& numbers)

  {

    sort(numbers.begin(), numbers.end());

  }
};

class numberInput
```

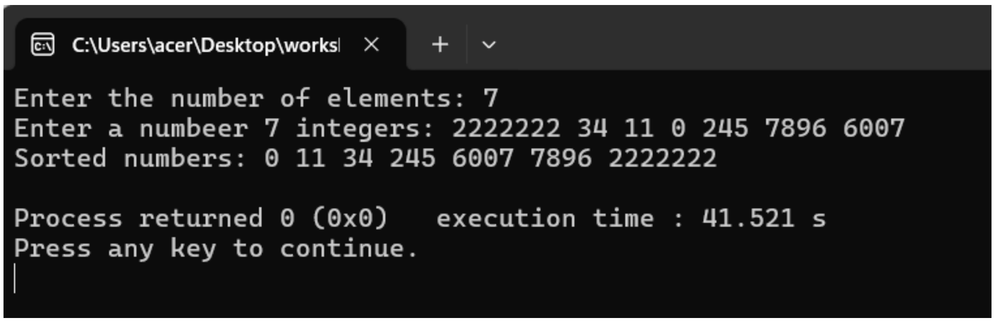
```
{
public:

   static vector<int> getNumbers()

   {
      int a;

      cout << "Enter the no. of elements: ";

      cin >> a;

      vector<int> numbers(a);

      cout << "Enter a numbeer " << a << " integers: ";


      for (int i = 0; i < a ; i++)

         {

         cin >> numbers[i];


         }

      return numbers;

   }
};

class NumberSorting


{

public:


   static void run()

   {
      vector<int> numbers = numberInput::getNumbers();

      NumberSorter::sortNumbers(numbers);

      cout << "Sorted numbers  in ascending order: ";

      for (int num : numbers)

         {

            cout << num << " ";

         }
```

```
        cout << endl;
    }
};

int main()
{
    NumberSorting::run();


    return 0;
}
```

OUTPUT:



```
C:\Users\acer\Desktop\worksl

Enter the number of elements: 5
Enter a numbeer 5 integers: 7 222 5673 0 11
Sorted numbers: 0 7 11 222 5673

Process returned 0 (0x0)   execution time : 10.900 s
Press any key to continue.
```



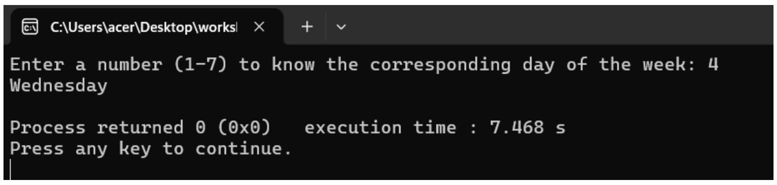
```
C:\Users\acer\Desktop\worksl

Enter the number of elements: 7
Enter a numbeer 7 integers: 2222222 34 11 0 245 7896 6007
Sorted numbers: 0 11 34 245 6007 7896 2222222

Process returned 0 (0x0)   execution time : 41.521 s
Press any key to continue.
```

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. [20 marks]

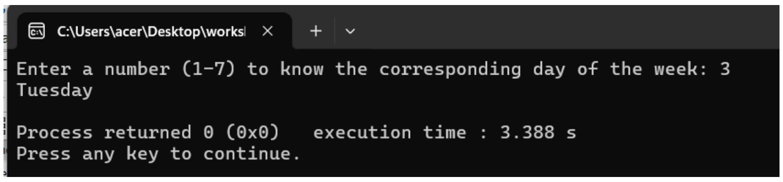
```
#include <iostream>


using namespace std;

class DayWeek

{

public:
   void getDayWeek(int day)


{
      switch (day)


      {

         case 1:

            cout << "Sunday" << endl;

            break;

         case 2:

            cout << "Monday" << endl;

            break;

         case 3:

            cout << "Tuesday" << endl;
            break;

         case 4:

            cout << "Wednesday" << endl;
            break;

         case 5:

            cout << "Thursday" << endl;
            break;

         case 6:

            cout << "Friday" << endl;
            break;


         case 7:

            cout << "Saturday" << endl;
            break;

         default:

            cout << "Your Input is Invalid. Please enter numbers between 1 and 7." << endl;

            break;

      }
   }
};
```

```
void getUserInput()

{
   int day;

   cout << "Enter a num for a day of the week: ";

   cin >> day;

   DayWeek W1;

   W1.getDayWeek(day);

}

int main()

{
   getUserInput();

   return 0;

}
```

OUTPUT:



```
Enter a number (1-7) to know the corresponding day of the week: 4
Wednesday

Process returned 0 (0x0)   execution time : 7.468 s
Press any key to continue.
```



```
Enter a number (1-7) to know the corresponding day of the week: 3
Tuesday

Process returned 0 (0x0)   execution time : 3.388 s
Press any key to continue.
```

Task 2: Programming Exercises:[Control Statements]

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:

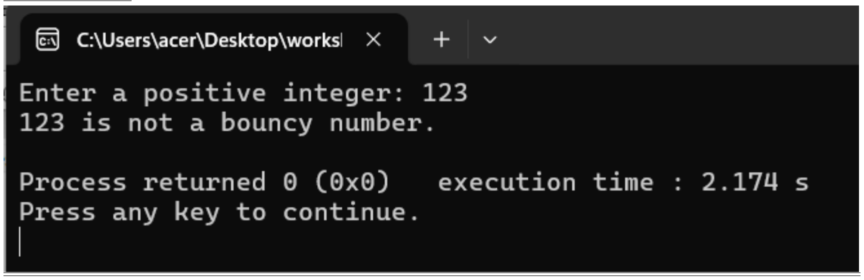
- **123 is NOT bouncy (digits consistently increase)**
- **321 is NOT bouncy (digits consistently decrease)**
- **120 is bouncy (neither consistently increasing nor decreasing)**

```
#include <iostream>

#include <vector>

using namespace std;
```

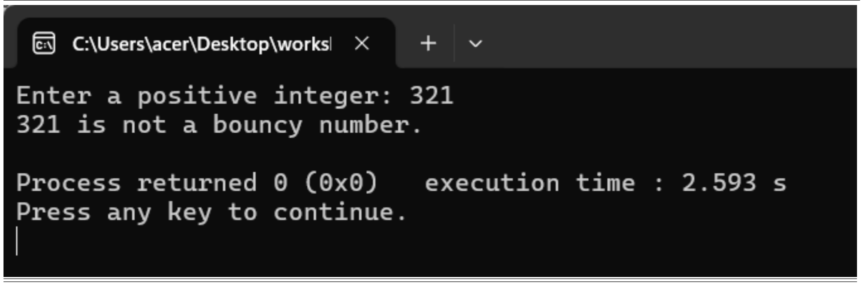
```
class BouncyNumber

{

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;
}
```

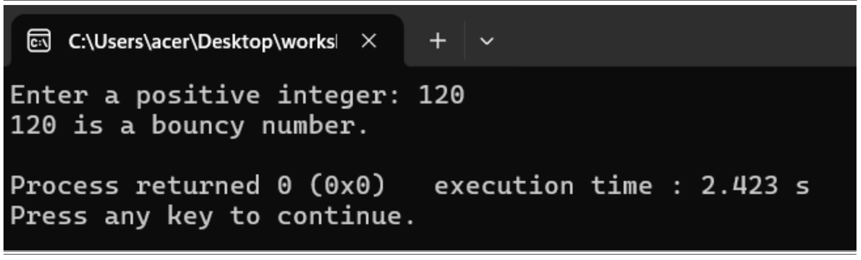
OUTPUT:



```
Enter a positive integer: 123
123 is not a bouncy number.

Process returned 0 (0x0)   execution time : 2.174 s
Press any key to continue.
```



```
Enter a positive integer: 321
321 is not a bouncy number.

Process returned 0 (0x0)   execution time : 2.593 s
Press any key to continue.
```



```
Enter a positive integer: 120
120 is a bouncy number.

Process returned 0 (0x0)   execution time : 2.423 s
Press any key to continue.
```

Task 3: Programming Exercises on Arrays

1. Write a program that manages a cinema ticket booking system. The program should display a 5x5 seating arrangement where: [25 marks]

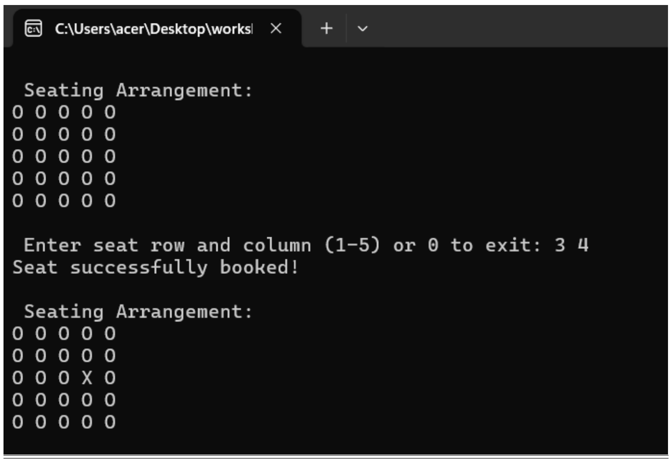
1. **Available seats are marked with 'O'**
2. **Booked seats are marked with 'X'**

```
#include <iostream>

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 << "\n Seating Arrangement:\n";

        for (int i = 0; i < 5; ++i)

            {

            for (int j = 0; j < 5; ++j)

            {

                cout << seats[i][j] << " ";

            }

            cout << endl;

        }
    }

    void bookSeat()

    {

        int row, col;

        while (true)

            {

            displaySeating();

            cout << "\n Enter seat row and column (1-5) or 0 to exit: ";

            cin >> row;

            if (row == 0)

                {
```

```
                cout << "Booking ended.\n";

                break;

            }

        cin >> col;


        row--;
        col--;

        if (row >= 0 && row < 5 && col >= 0 && col < 5)

            {

            if (seats[row][col] == 'O')

                {

                    seats[row][col] = 'X';


                    cout << "Seat successfully booked!\n";

                }

             else

                {

                cout << "Seat already booked. Try another one.\n";

            }

        }

            else

                {

                    cout << "Invalid input. Please choose between 1 and 5.\n";

        }
      }
   }
};

int main()

{
   Cinema c1;

   c1.bookSeat();

   return 0;
```

```
}
```

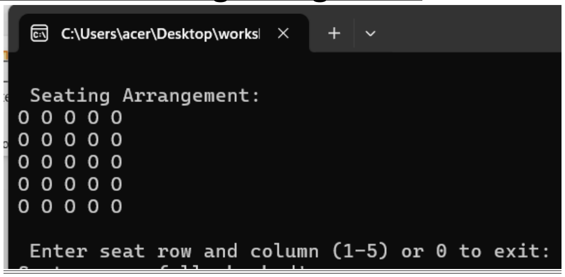
<u>OUTPUT:</u>



```
C:\Users\acer\Desktop\worksl

 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 seat row and column (1-5) or 0 to exit: 3 4
Seat successfully booked!

 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
```

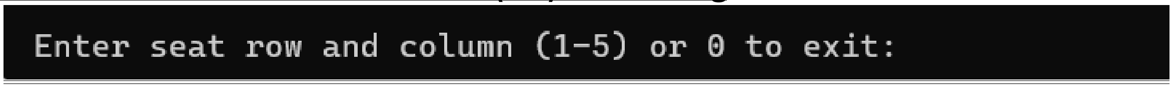
<u>Program should:</u>

<u>1.Display the current seating arrangement</u>



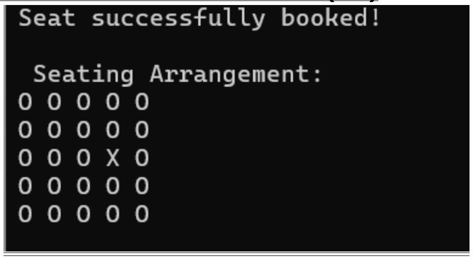
```
C:\Users\acer\Desktop\worksl

 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 seat row and column (1-5) or 0 to exit:
```

2. **<u>Ask user for row and column number (1-5) for booking</u>**



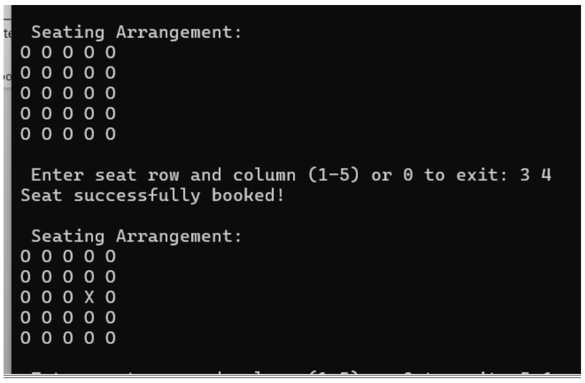
```
 Enter seat row and column (1-5) or 0 to exit:
```

3. **<u>Mark that seat as booked ('X')</u>**



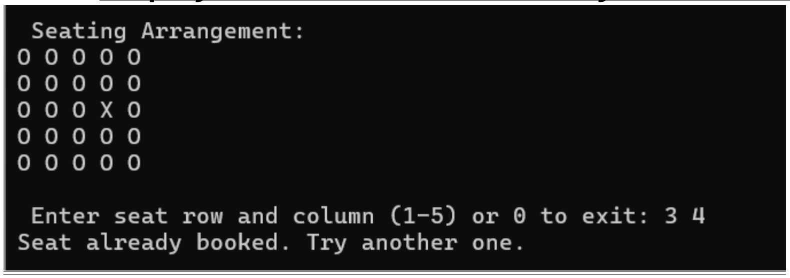
```
Seat successfully booked!

 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
```

4. **<u>Show updated seating after each booking</u>**



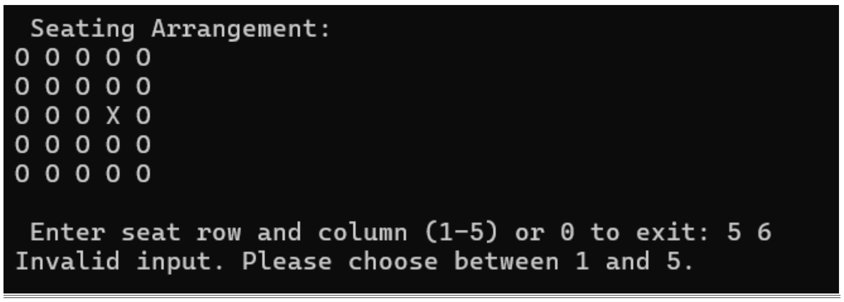
```
 Seating Arrangement:
O O O O O
O O O O O
O O O O O
O O O O O
O O O O O

 Enter seat row and column (1-5) or 0 to exit: 3 4
Seat successfully booked!

 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
```

5. **Display error if user selects already booked seat**



```
 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 seat row and column (1-5) or 0 to exit: 3 4
Seat already booked. Try another one.
```

6. **Display error if user enters invalid row/column numbers**



```
 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 seat row and column (1-5) or 0 to exit: 5 6
Invalid input. Please choose between 1 and 5.
```