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.

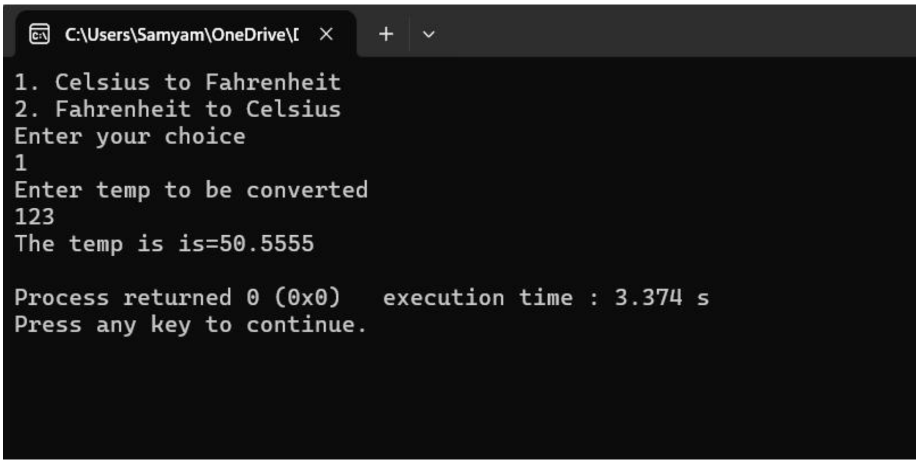
> Use appropriate data types for temperature and handle error like non-numeric input.
> Use the following formula for conversion:
> F = (C x 9/5) + 32
> C = (F - 32) x 5/9 **[10 marks]**

```
#include<iostream>
using namespace std;
class Temperature
{
    float choice,C,F;
public:
    void menu()
    {
        cout<<"1. Celsius to fahrenheit"<<endl;
        cout<<"2. Fahrenheit to Celsius"<<endl;
        cout<<"ENTER YOUR CHOICE"<<endl;
        cin>>choice;
    }
    void input()
    {
        if(choice==1)
        {
            cout<<"enter the temperature to be converted to
fahrenheit"<<endl;
            cin>>C;
            calculation1();
        }
        else if(choice==2)
        {
            cout<<"enter rhe temperature to be converted to Celsius"<<endl;
            cin>>F;
            calculation2();
        }
        else
        {
            cout<<"ERROR"<<endl;

        }}
```

```
void calculation1()
{
      F=1.8*C+32;
}

void calculation2()
{
   C=(F*9/5)+32;
}
void display()
{
     if(choice==1)
     {
        cout<<"the tempreture is" <<F<<endl;
     }
     else if(choice==2)
     {
         cout<<"the tempreture is "<<C<<endl;
     }
}
```

```
};


    int main()
    {
        Temperature T1;
        T1.menu();
        T1.input();
        T1.display();
        return 0;
    }
```

<u>OUTPUT</u>



```
1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
Enter your choice
1
Enter temp to be converted
123
The temp is is=50.5555

Process returned 0 (0x0)   execution time : 3.374 s
Press any key to continue.
```

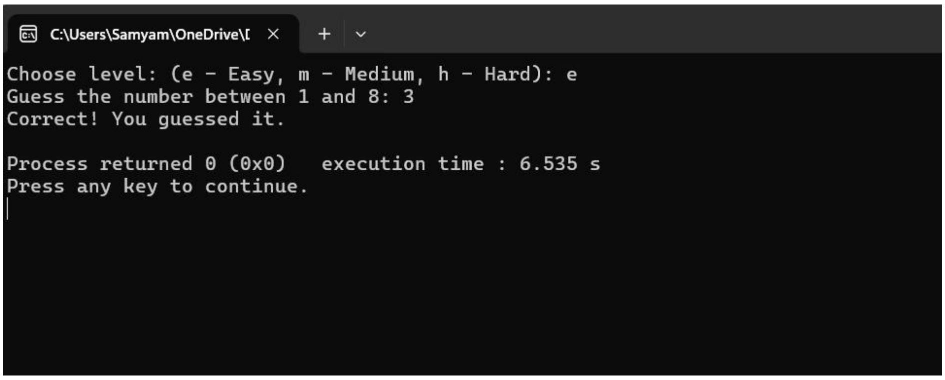
Write a C++ program to implement a number guessing game with different difficulty levels.Easy difficulty ranges from 1-8, medium from 1-30, hard from 1-50.Then,generate a random number to check if the guess is correct based on the user's selection.

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

class NumberGuessingGame {
private:
    int number;
    int guess;

public:
    void playGame() {
        char level;
        int range;
```

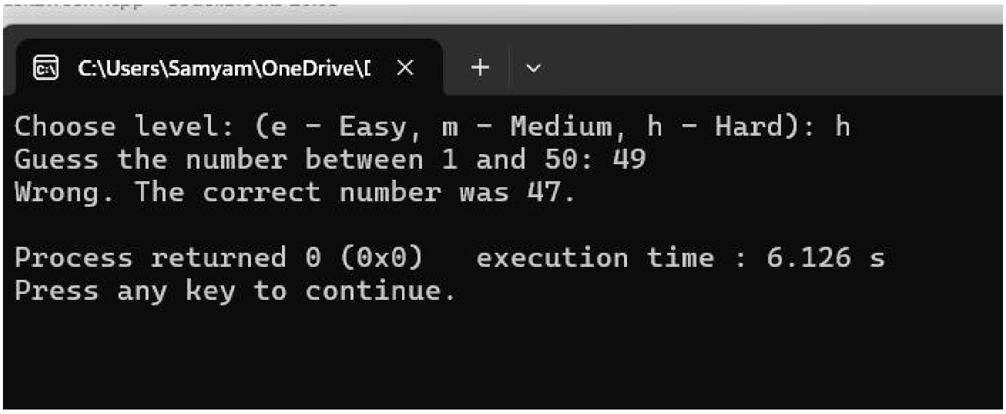
```
cout << "Choose level: (e - Easy, m - Medium, h - Hard): ";
cin >> level;

if (level == 'e') {
    range = 8;
} else if (level == 'm') {
    range = 30;
} else if (level == 'h') {
    range = 50;
} else {
    cout << "Invalid level!" << endl;
    return;
}

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

cout << "Guess the number between 1 and " << range << ": ";
cin >> guess;

if (guess == number) {
```

```
            cout << "Correct! You guessed it." << endl;
        } else {
            cout << "Wrong. The correct number was " << number << "." <<
endl;
        }
    }
};

int main() {
    NumberGuessingGame game;
    game.playGame();
    return 0;
}
```

OUTPUT



```
Choose level: (e - Easy, m - Medium, h - Hard): e
Guess the number between 1 and 8: 3
Correct! You guessed it.

Process returned 0 (0x0)   execution time : 6.535 s
Press any key to continue.
```

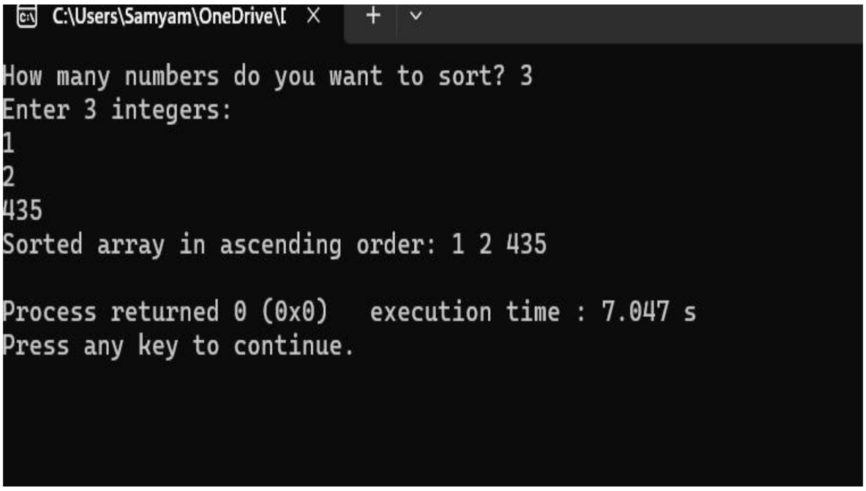


```
Choose level: (e - Easy, m - Medium, h - Hard): h
Guess the number between 1 and 50: 49
Wrong. The correct number was 47.

Process returned 0 (0x0)   execution time : 6.126 s
Press any key to continue.
```

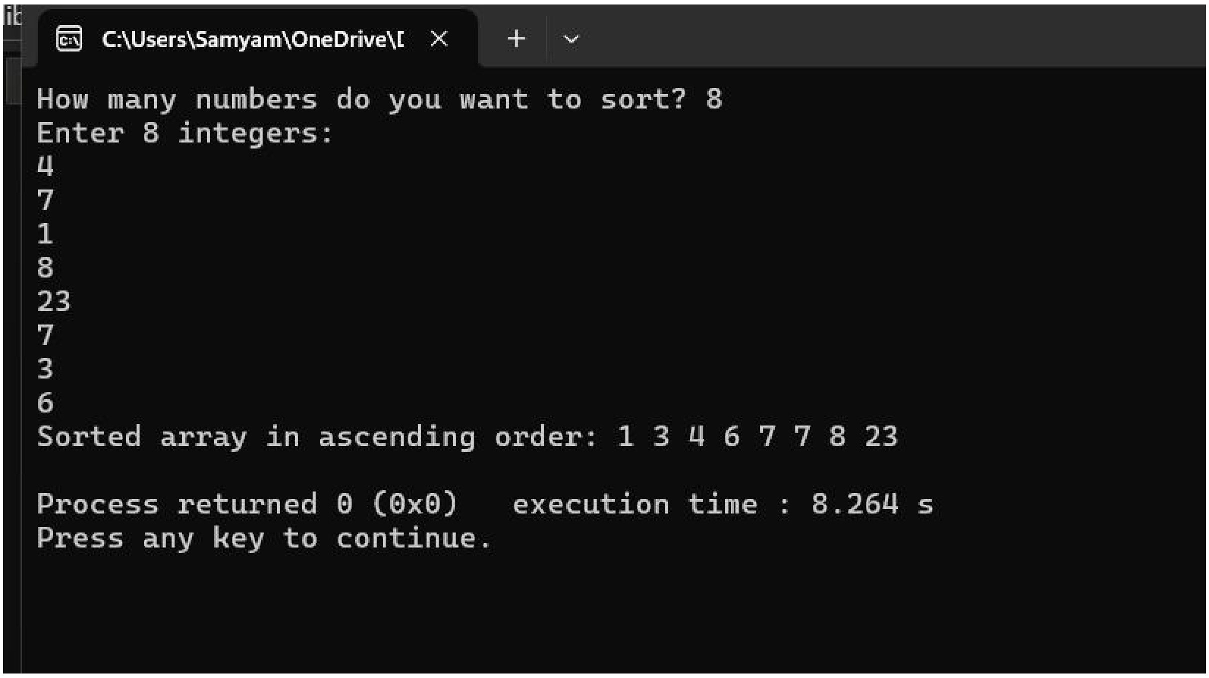
Write a program that reads an array of integer numbers from the user and sorts the numbers in the ascending order.

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

class ArraySorter {
private:
   vector<int> numbers;

public:
   void inputNumbers() {
      int n, value;
      cout << "How many numbers do you want to sort? ";
      cin >> n;

      cout << "Enter " << n << " integers:" << endl;
      for (int i = 0; i < n; ++i) {
            cin >> value;
            numbers.push_back(value);
        }
    }

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

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

int main() {
    ArraySorter sorter;
    sorter.inputNumbers();
    sorter.sortArray();
    sorter.displaySorted();
    return 0;
}
```

OUTPUT



```
How many numbers do you want to sort? 3
Enter 3 integers:
1
2
435
Sorted array in ascending order: 1 2 435

Process returned 0 (0x0)   execution time : 7.047 s
Press any key to continue.
```

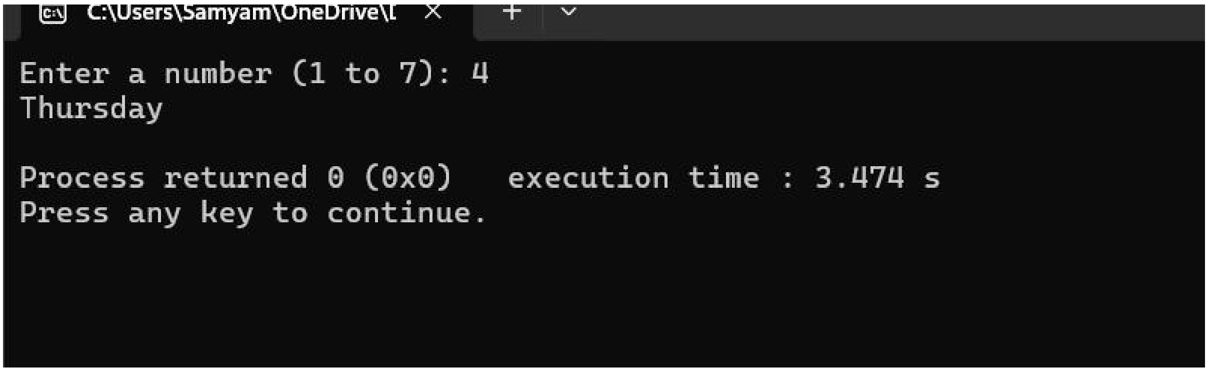


```
How many numbers do you want to sort? 8
Enter 8 integers:
4
7
1
8
23
7
3
6
Sorted array in ascending order: 1 3 4 6 7 7 8 23

Process returned 0 (0x0)   execution time : 8.264 s
Press any key to continue.
```

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

```
#include <iostream>

using namespace std;



class DayOfWeek {

private:

    int dayNumber;



public:

    void getInput() {

        cout << "Enter a number (1 to 7): ";
        cin >> dayNumber;
    }

    void showDay() {
        switch (dayNumber) {
            case 1: cout << "Monday" << endl; break;
            case 2: cout << "Tuesday" << endl; break;
            case 3: cout << "Wednesday" << endl; break;
            case 4: cout << "Thursday" << endl; break;
            case 5: cout << "Friday" << endl; break;
            case 6: cout << "Saturday" << endl; break;
            case 7: cout << "Sunday" << endl; break;
            default: cout << "Invalid number. Please enter 1 to 7." << endl;
        }
    }
};

int main() {
    DayOfWeek day;
    day.getInput();
    day.showDay();
    return 0;

}
```

OUTPUT



C:\Users\Samyam\OneDrive\[

```
Enter a number (1 to 7): 4
Thursday

Process returned 0 (0x0)   execution time : 3.474 s
Press any key to continue.
```

Task 2: Programming Exercises:[Control Statements]

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:

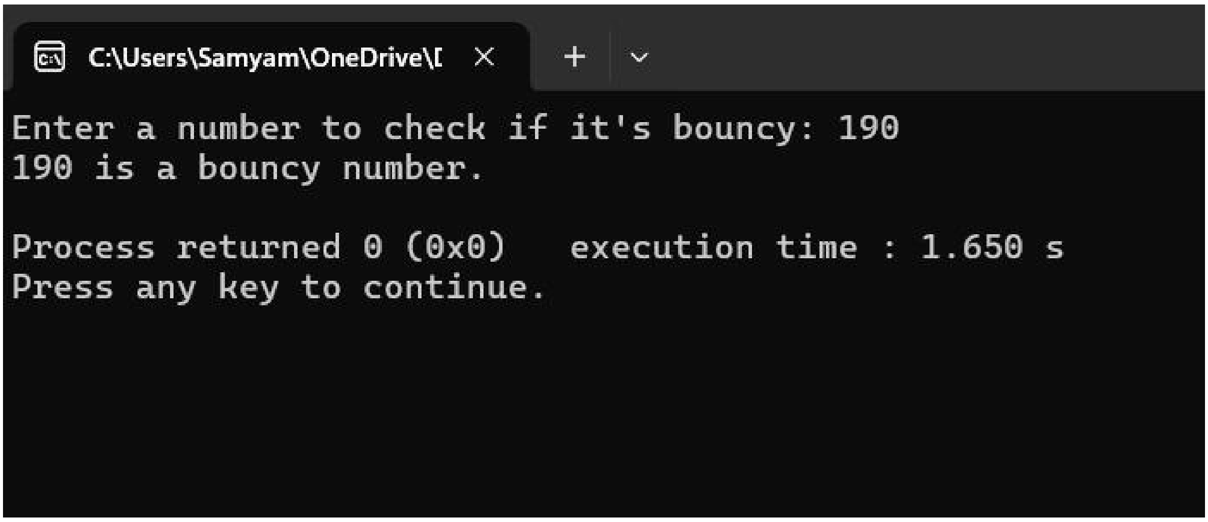
123 is NOT bouncy (digits consistently increase)
321 is NOT bouncy (digits consistently decrease)
120 is bouncy (neither consistently increasing nor decreasing)

```
#include <iostream>

using namespace std;



class BouncyNumber {

private:

    int number;



public:

    void getInput() {
```

```
    cout << "Enter a number to check if it's bouncy: ";
    cin >> number;
}

bool isBouncy() {
    bool increasing = false;
    bool decreasing = false;

    int lastDigit = number % 10;
    int num = number / 10;

    while (num > 0) {
        int currentDigit = num % 10;
        if (currentDigit < lastDigit)
            increasing = true;
        else if (currentDigit > lastDigit)
            decreasing = true;

        lastDigit = currentDigit;
        num /= 10;
    }
```

```
        return increasing && decreasing;
    }

    void showResult() {
        if (number < 100) {
            cout << "Not a bouncy number (less than 100)." << endl;
        } else if (isBouncy()) {
            cout << number << " is a bouncy number." << endl;
        } else {
            cout << number << " is not a bouncy number." << endl;
        }
    }
};

int main() {
    BouncyNumber bn;
    bn.getInput();
    bn.showResult();
    return 0;
}
```

OUTPUT



```
Enter a number to check if it's bouncy: 190
190 is a bouncy number.

Process returned 0 (0x0)   execution time : 1.650 s
Press any key to continue.
```

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

Available seats are marked with 'O'

Booked seats are marked with 'X'

Program should:

Display the current seating arrangement

Ask user for row and column number (1-5) for booking

Mark that seat as booked ('X')

Show updated seating after each booking

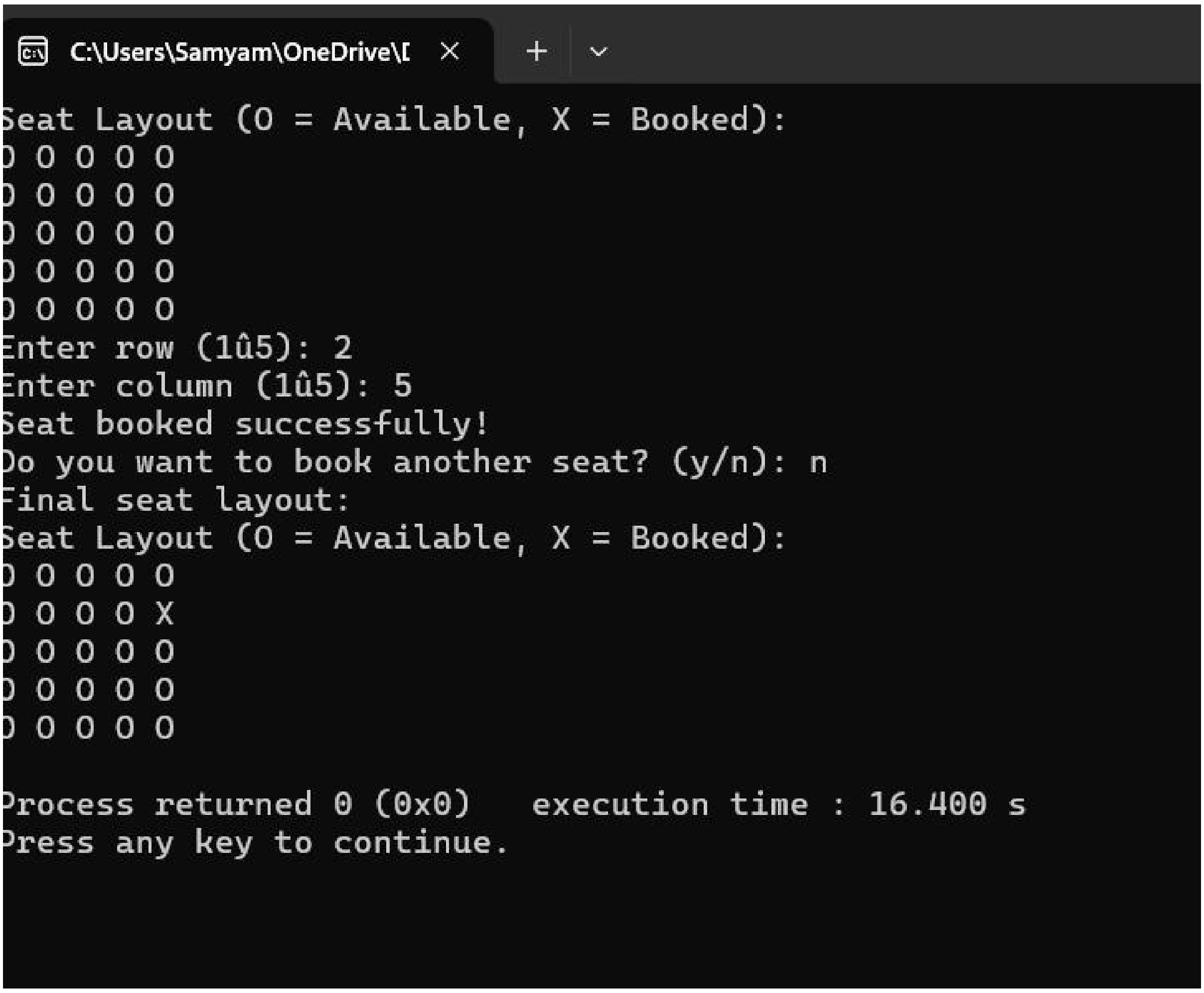
Display error if user selects already booked seat

Display error if user enters invalid row/column numbers

```
#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 displaySeats() {
        cout << "Seat Layout (O = Available, X = Booked):¥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;
        cout << "Enter row (1–5): ";
        cin >> row;
        cout << "Enter column (1–5): ";
        cin >> col;

        if (row < 1 || row > 5 || col < 1 || col > 5) {
            cout << "Invalid seat position!" << endl;
        } else if (seats[row - 1][col - 1] == 'X') {
            cout << "Seat already booked!" << endl;
        } else {
            seats[row - 1][col - 1] = 'X';
            cout << "Seat booked successfully!" << endl;
        }
    }
};

int main() {
    Cinema cinema;
    char choice;

    do {
        cinema.displaySeats();
        cinema.bookSeat();
        cout << "Do you want to book another seat? (y/n): ";
        cin >> choice;
    } while (choice == 'y' || choice == 'Y');

    cout << "Final seat layout:¥n";
    cinema.displaySeats();

    return 0;
}
```

OUTPUT



```
Seat Layout (O = Available, X = Booked):
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 (1û5): 2
Enter column (1û5): 5
Seat booked successfully!
Do you want to book another seat? (y/n): n
Final seat layout:
Seat Layout (O = Available, X = Booked):
O O O O O
O O O O X
O O O O O
O O O O O
O O O O O

Process returned 0 (0x0)   execution time : 16.400 s
Press any key to continue.
```