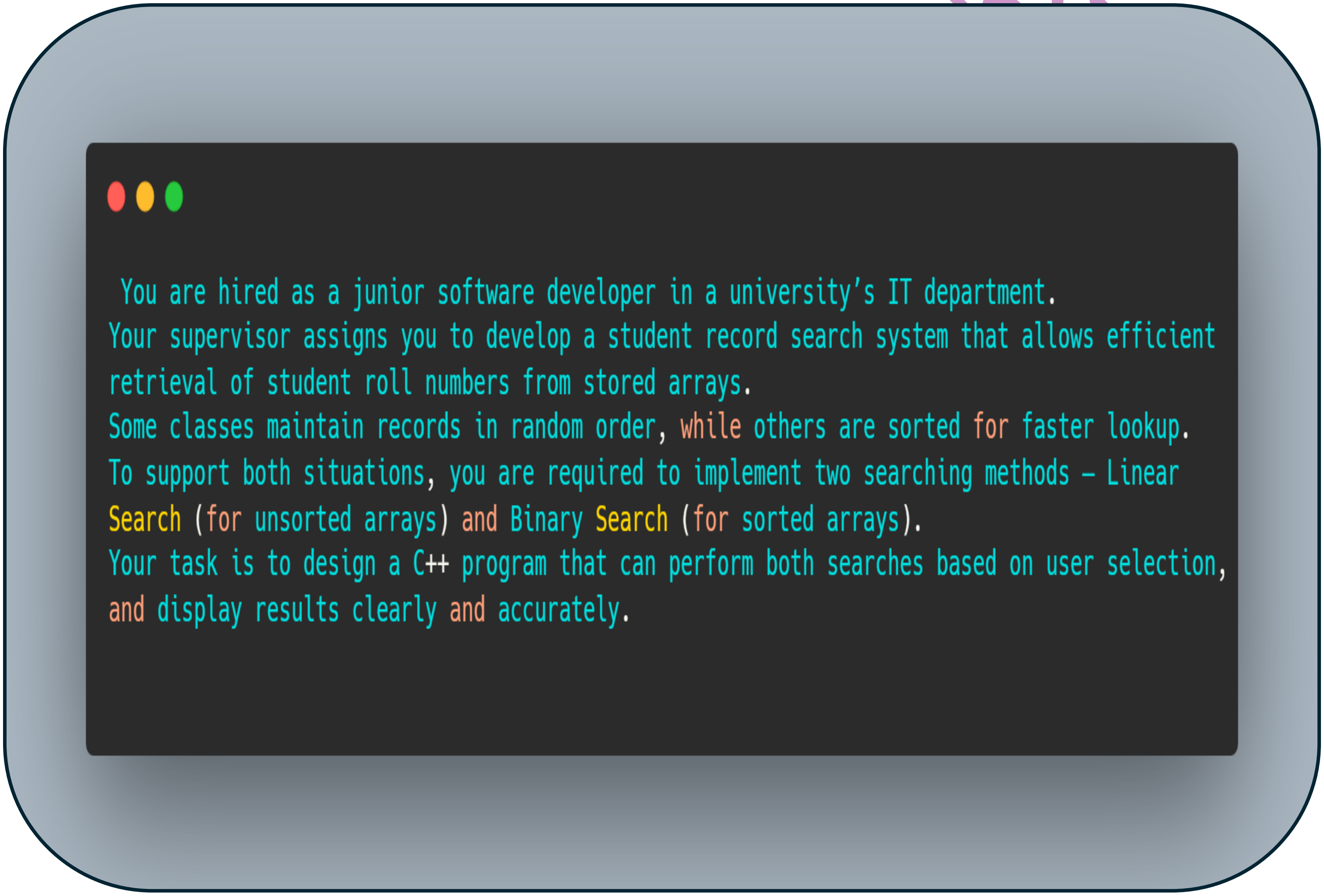
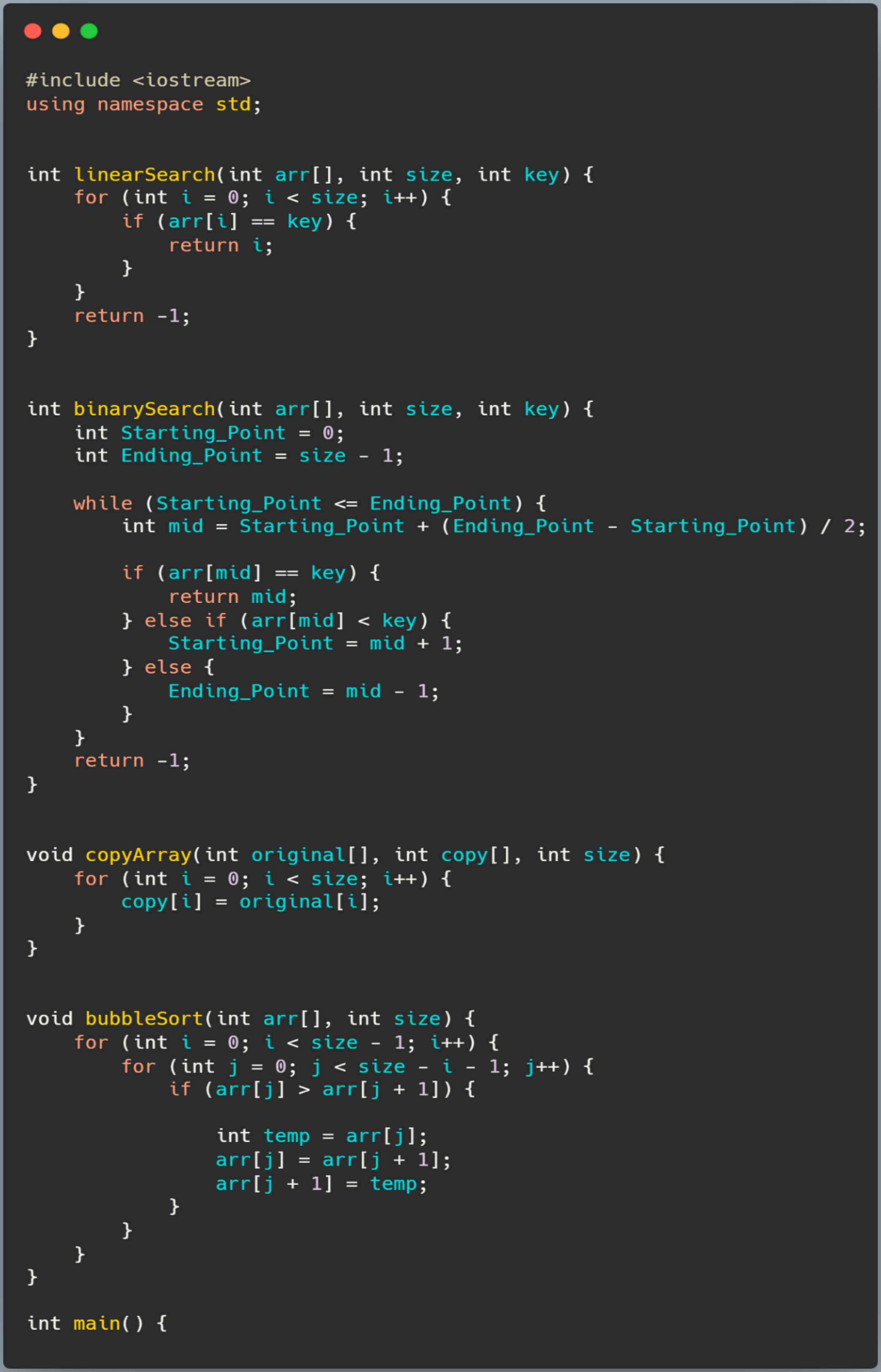
Task 5 [student record search system]

Task:

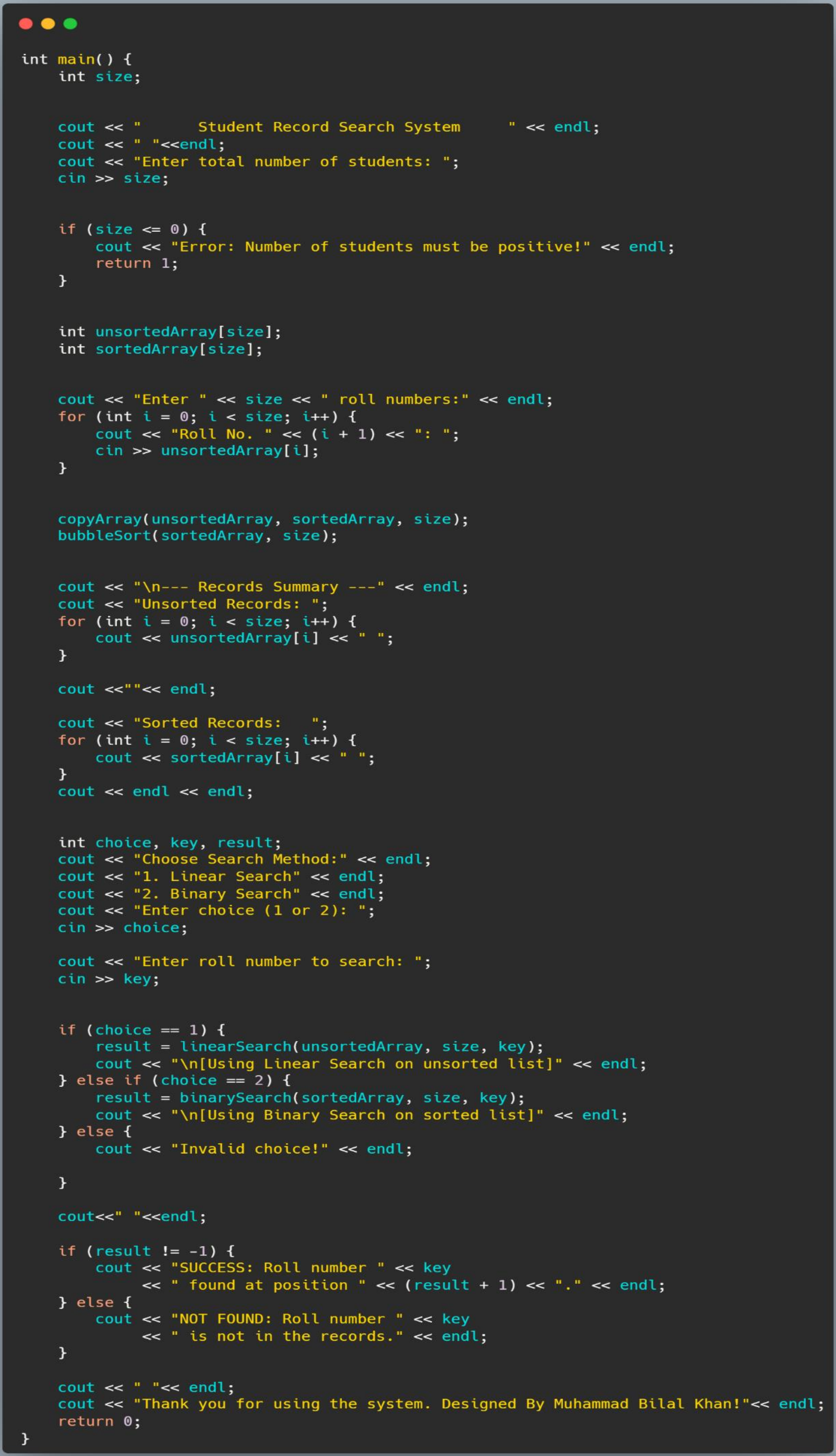


You are hired as a junior software developer in a university's IT department.
Your supervisor assigns you to develop a student record search system that allows efficient retrieval of student roll numbers from stored arrays.
Some classes maintain records in random order, while others are sorted for faster lookup.
To support both situations, you are required to implement two searching methods – Linear Search (for unsorted arrays) and Binary Search (for sorted arrays).
Your task is to design a C++ program that can perform both searches based on user selection, and display results clearly and accurately.

Code:



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


int linearSearch(int arr[], int size, int key) {
    for (int i = 0; i < size; i++) {
        if (arr[i] == key) {
            return i;
        }
    }
    return -1;
}


int binarySearch(int arr[], int size, int key) {
    int Starting_Point = 0;
    int Ending_Point = size - 1;

    while (Starting_Point <= Ending_Point) {
        int mid = Starting_Point + (Ending_Point - Starting_Point) / 2;

        if (arr[mid] == key) {
            return mid;
        } else if (arr[mid] < key) {
            Starting_Point = mid + 1;
        } else {
            Ending_Point = mid - 1;
        }
    }
    return -1;
}


void copyArray(int original[], int copy[], int size) {
    for (int i = 0; i < size; i++) {
        copy[i] = original[i];
    }
}


void bubbleSort(int arr[], int size) {
    for (int i = 0; i < size - 1; i++) {
        for (int j = 0; j < size - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {

                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

int main() {
```



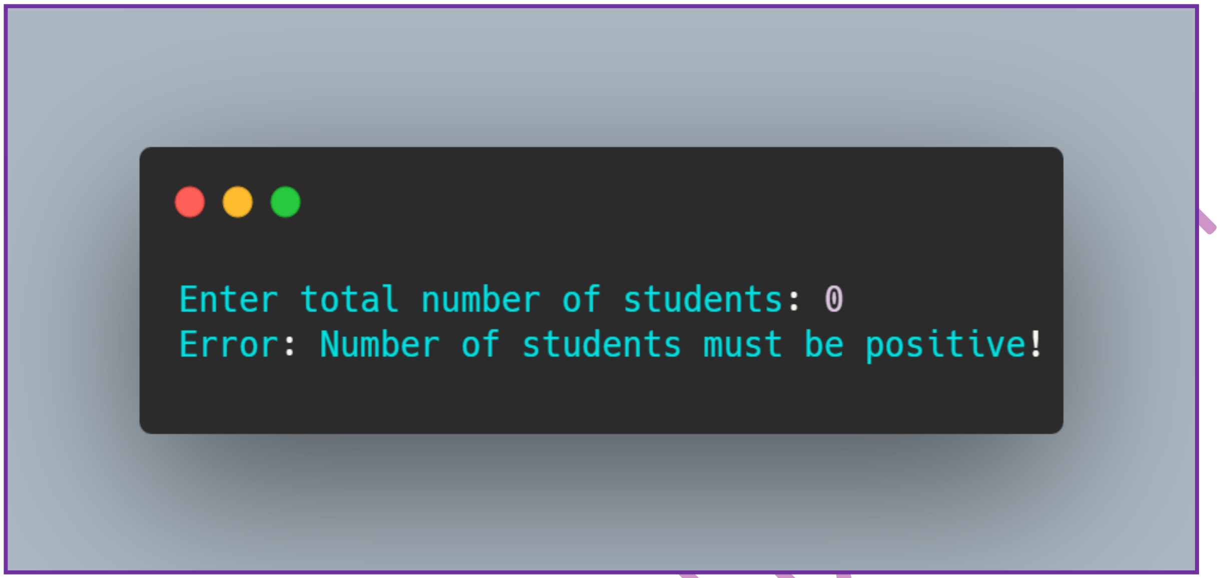
```cpp
int main() {
    int size;


    cout << "      Student Record Search System     " << endl;
    cout << " "<<endl;
    cout << "Enter total number of students: ";
    cin >> size;


    if (size <= 0) {
        cout << "Error: Number of students must be positive!" << endl;
        return 1;
    }


    int unsortedArray[size];
    int sortedArray[size];


    cout << "Enter " << size << " roll numbers:" << endl;
    for (int i = 0; i < size; i++) {
        cout << "Roll No. " << (i + 1) << ": ";
        cin >> unsortedArray[i];
    }


    copyArray(unsortedArray, sortedArray, size);
    bubbleSort(sortedArray, size);


    cout << "\n--- Records Summary ---" << endl;
    cout << "Unsorted Records: ";
    for (int i = 0; i < size; i++) {
        cout << unsortedArray[i] << " ";
    }

    cout <<""<< endl;

    cout << "Sorted Records:   ";
    for (int i = 0; i < size; i++) {
        cout << sortedArray[i] << " ";
    }
    cout << endl << endl;


    int choice, key, result;
    cout << "Choose Search Method:" << endl;
    cout << "1. Linear Search" << endl;
    cout << "2. Binary Search" << endl;
    cout << "Enter choice (1 or 2): ";
    cin >> choice;

    cout << "Enter roll number to search: ";
    cin >> key;


    if (choice == 1) {
        result = linearSearch(unsortedArray, size, key);
        cout << "\n[Using Linear Search on unsorted list]" << endl;
    } else if (choice == 2) {
        result = binarySearch(sortedArray, size, key);
        cout << "\n[Using Binary Search on sorted list]" << endl;
    } else {
        cout << "Invalid choice!" << endl;

    }

    cout<<" "<<endl;

    if (result != -1) {
        cout << "SUCCESS: Roll number " << key
             << " found at position " << (result + 1) << "." << endl;
    } else {
        cout << "NOT FOUND: Roll number " << key
             << " is not in the records." << endl;
    }

    cout << " "<< endl;
    cout << "Thank you for using the system. Designed By Muhammad Bilal Khan!"<< endl;
    return 0;
}
```

Table Of Output:

Output	Condition
Error message and program terminates	Invalid size [size <= 0]
SUCCESS message (linear search)	Linear Search – Found [choice == 1 and key exists]
NOT FOUND message (linear search)	Linear Search – Not Found [choice == 1 and key does not exist]
SUCCESS message (binary search)	Binary Search – Found [choice == 2 and key exists]
NOT FOUND message (binary search)	Binary Search – Not Found [choice == 2 and key does not exist]
Invalid choice!	Invalid Search Choice [choice != 1 && choice != 2]

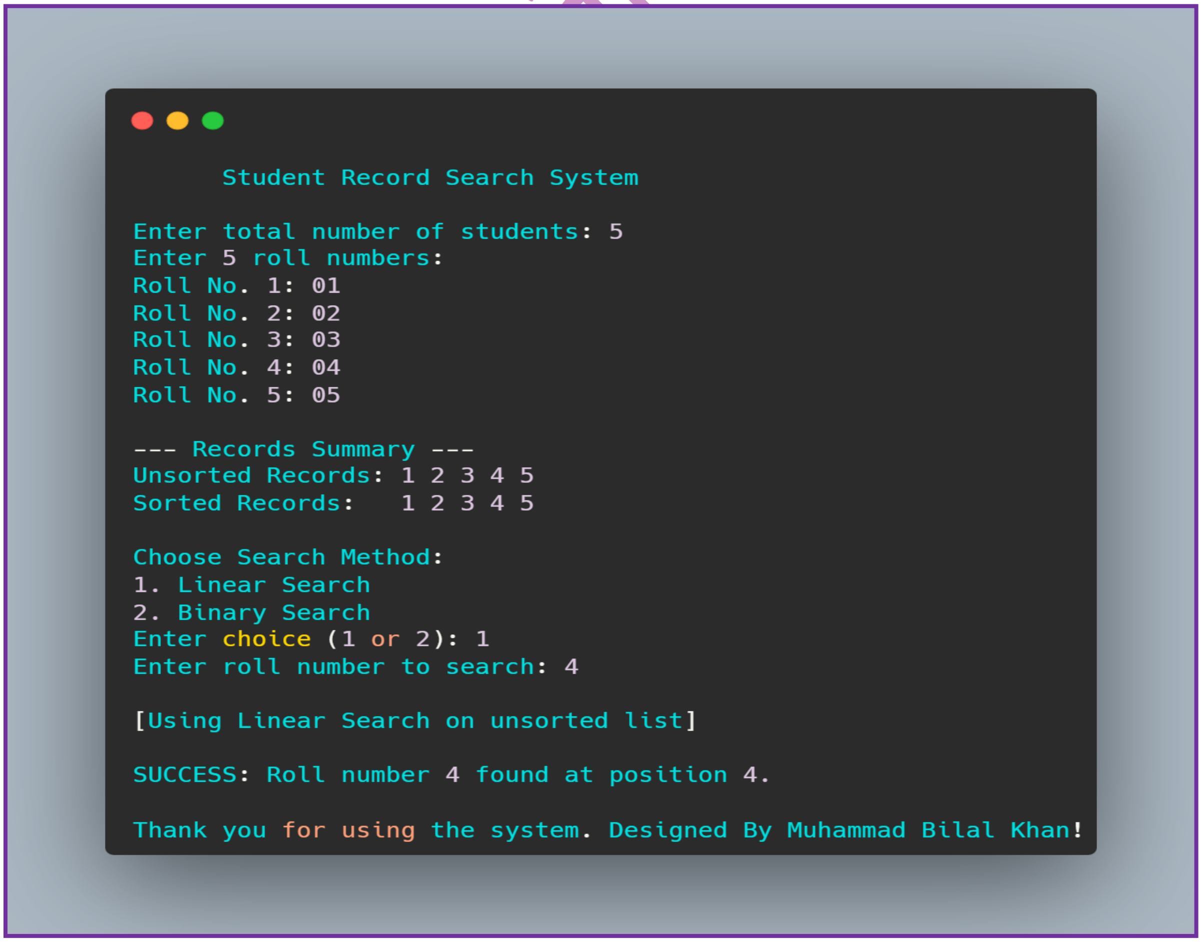
Total 6 Possible Outputs

When Invalid size [size <= 0] :



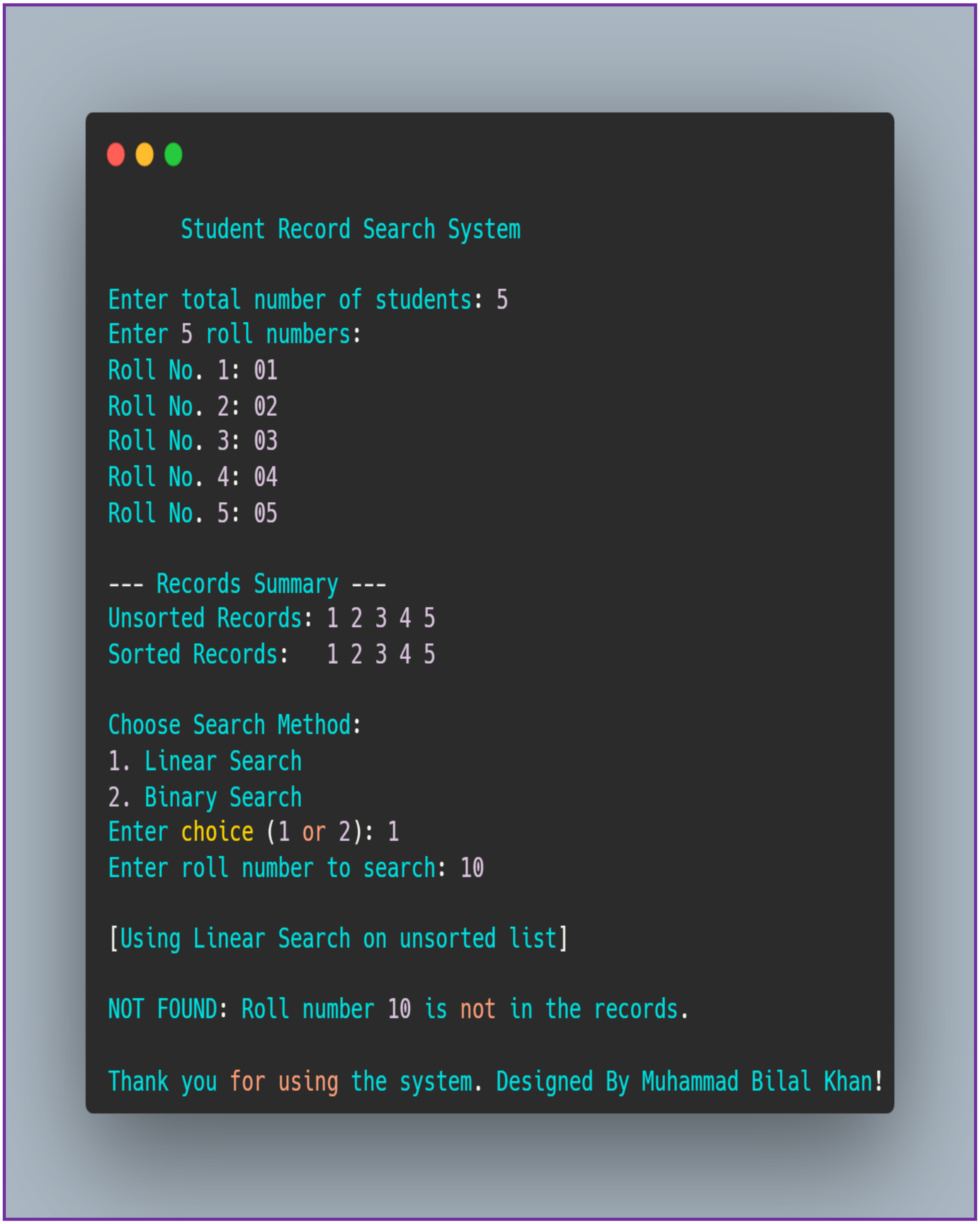
```
Enter total number of students: 0
Error: Number of students must be positive!
```

When Linear Search – Found :



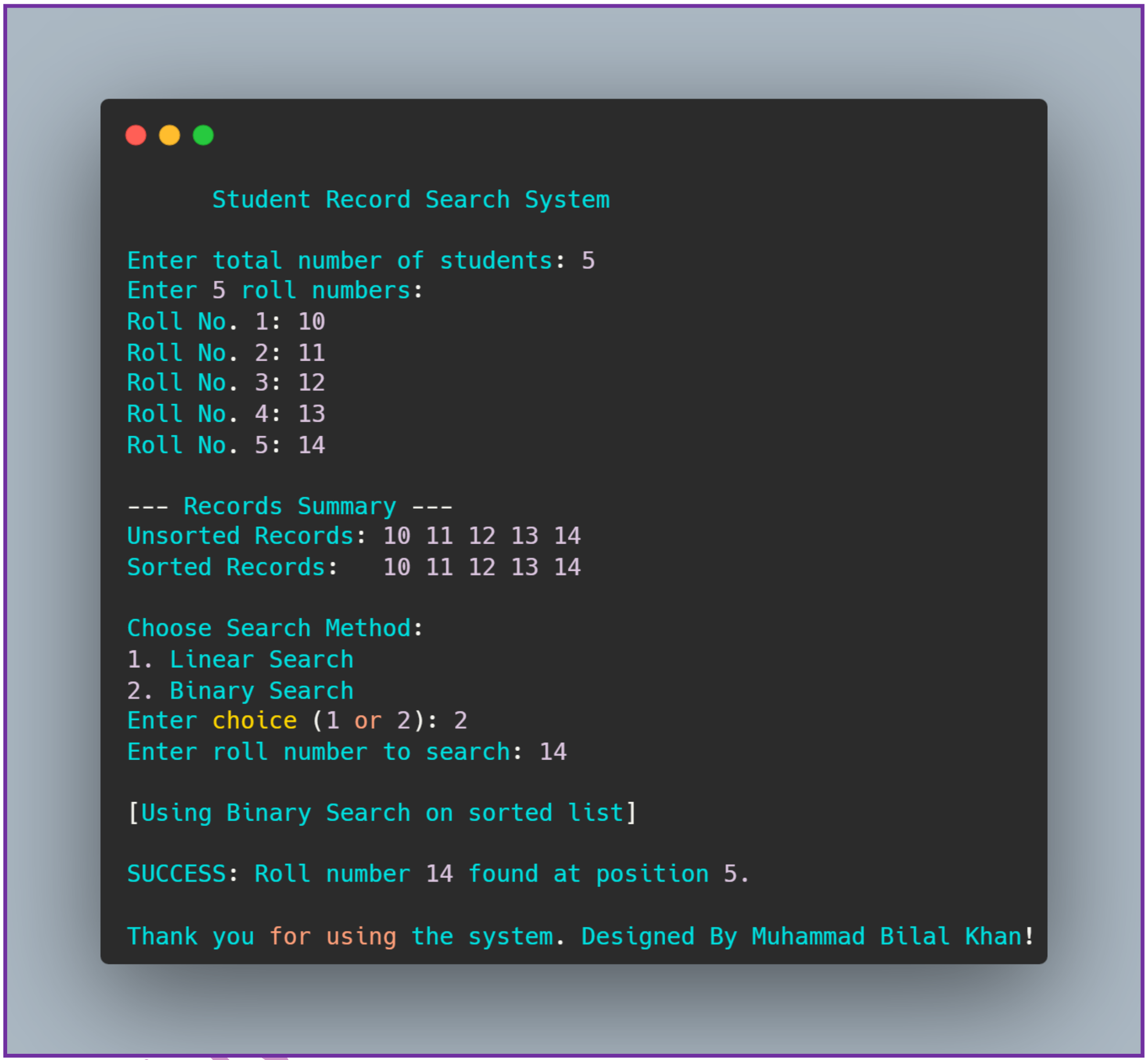
```
      Student Record Search System

Enter total number of students: 5
Enter 5 roll numbers:
Roll No. 1: 01
Roll No. 2: 02
Roll No. 3: 03
Roll No. 4: 04
Roll No. 5: 05

--- Records Summary ---
Unsorted Records: 1 2 3 4 5
Sorted Records:   1 2 3 4 5

Choose Search Method:
1. Linear Search
2. Binary Search
Enter choice (1 or 2): 1
Enter roll number to search: 4

[Using Linear Search on unsorted list]

SUCCESS: Roll number 4 found at position 4.

Thank you for using the system. Designed By Muhammad Bilal Khan!
```

When Linear Search – Not Found :



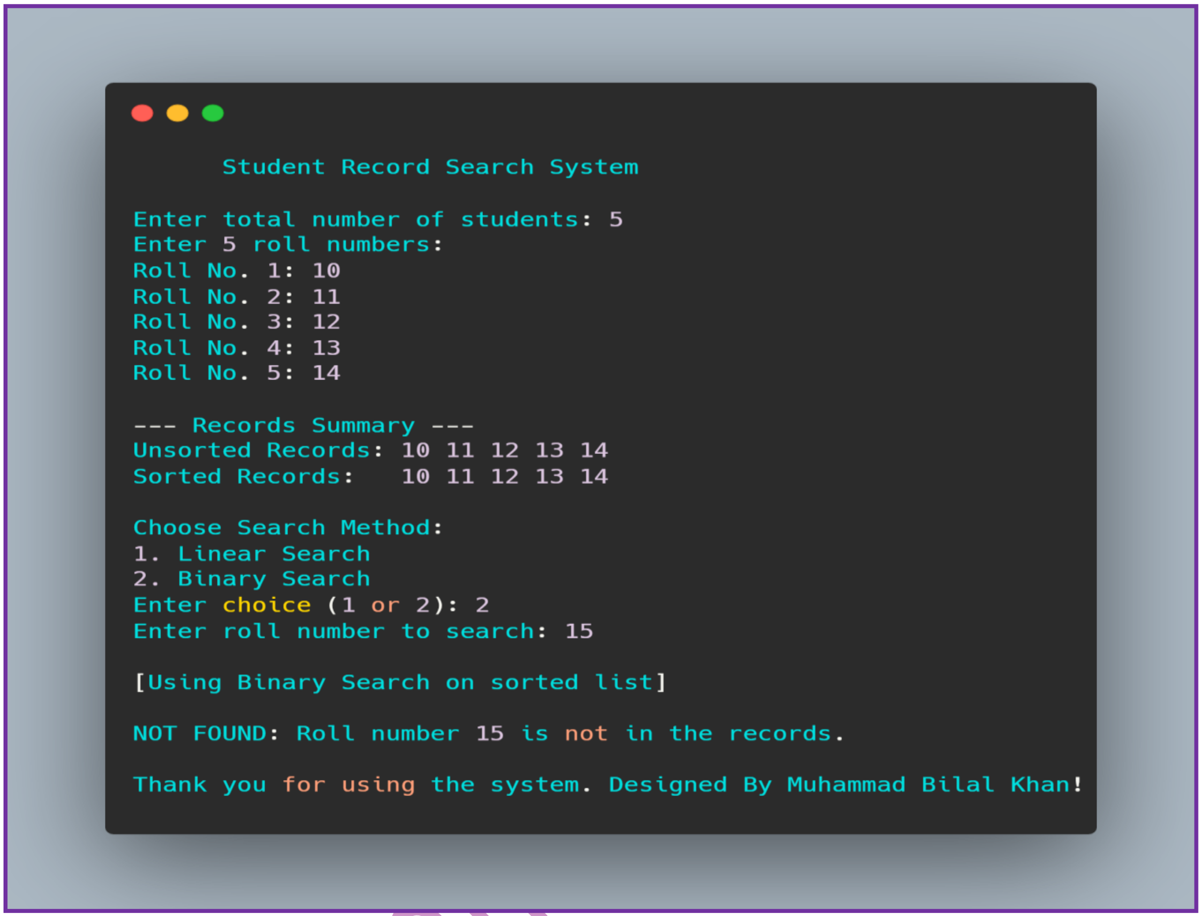
```
      Student Record Search System

Enter total number of students: 5
Enter 5 roll numbers:
Roll No. 1: 01
Roll No. 2: 02
Roll No. 3: 03
Roll No. 4: 04
Roll No. 5: 05

--- Records Summary ---
Unsorted Records: 1 2 3 4 5
Sorted Records:   1 2 3 4 5

Choose Search Method:
1. Linear Search
2. Binary Search
Enter choice (1 or 2): 1
Enter roll number to search: 10

[Using Linear Search on unsorted list]

NOT FOUND: Roll number 10 is not in the records.

Thank you for using the system. Designed By Muhammad Bilal Khan!
```

When Binary Search – Found :



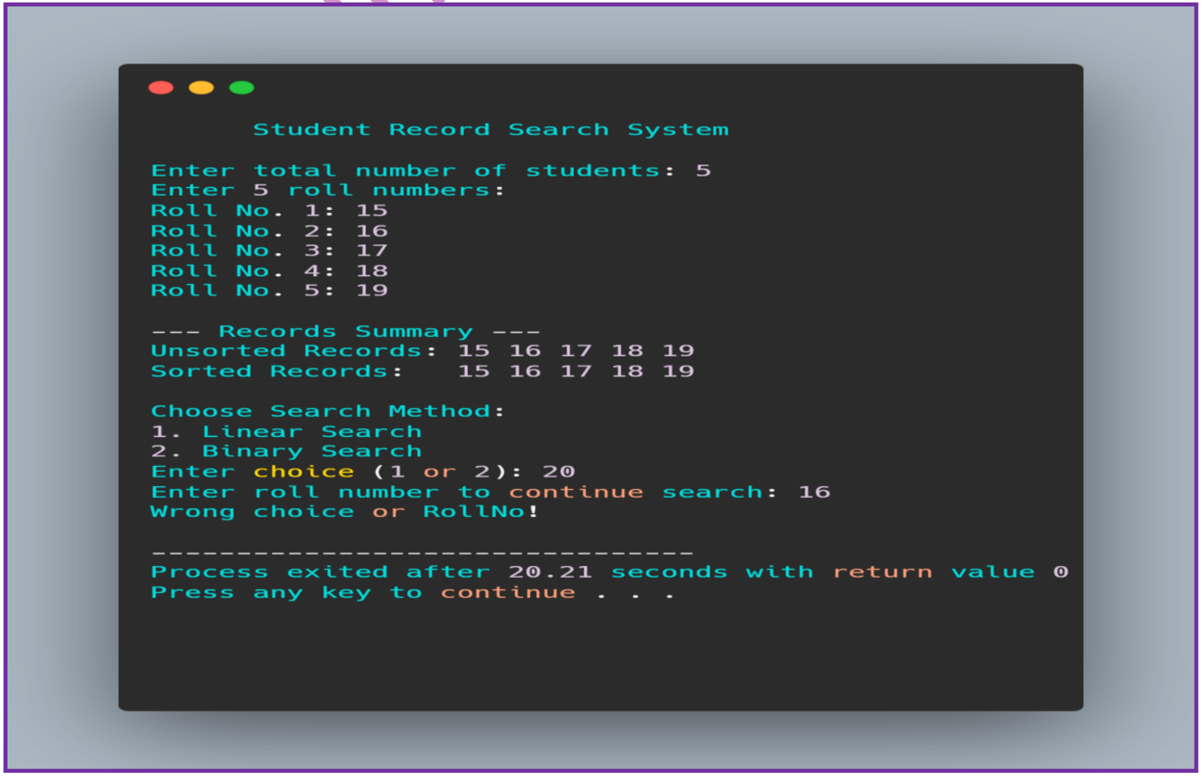
```
      Student Record Search System

Enter total number of students: 5
Enter 5 roll numbers:
Roll No. 1: 10
Roll No. 2: 11
Roll No. 3: 12
Roll No. 4: 13
Roll No. 5: 14

--- Records Summary ---
Unsorted Records: 10 11 12 13 14
Sorted Records:   10 11 12 13 14

Choose Search Method:
1. Linear Search
2. Binary Search
Enter choice (1 or 2): 2
Enter roll number to search: 14

[Using Binary Search on sorted list]

SUCCESS: Roll number 14 found at position 5.

Thank you for using the system. Designed By Muhammad Bilal Khan!
```

When Binary Search – Not Found :



```
      Student Record Search System

Enter total number of students: 5
Enter 5 roll numbers:
Roll No. 1: 10
Roll No. 2: 11
Roll No. 3: 12
Roll No. 4: 13
Roll No. 5: 14

--- Records Summary ---
Unsorted Records: 10 11 12 13 14
Sorted Records:   10 11 12 13 14

Choose Search Method:
1. Linear Search
2. Binary Search
Enter choice (1 or 2): 2
Enter roll number to search: 15

[Using Binary Search on sorted list]

NOT FOUND: Roll number 15 is not in the records.

Thank you for using the system. Designed By Muhammad Bilal Khan!
```

When Invalid Search Choice :



```
      Student Record Search System

Enter total number of students: 5
Enter 5 roll numbers:
Roll No. 1: 15
Roll No. 2: 16
Roll No. 3: 17
Roll No. 4: 18
Roll No. 5: 19

--- Records Summary ---
Unsorted Records: 15 16 17 18 19
Sorted Records:   15 16 17 18 19

Choose Search Method:
1. Linear Search
2. Binary Search
Enter choice (1 or 2): 20
Enter roll number to continue search: 16
Wrong choice or RollNo!

--------------------------------
Process exited after 20.21 seconds with return value 0
Press any key to continue . . .
```

Click here to Get this code on GitHub

Click here to Test this Code by Yourself.