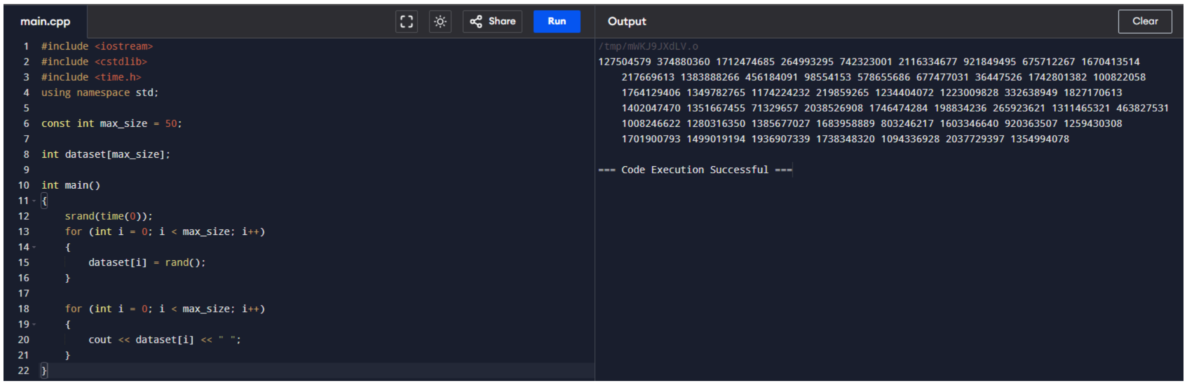
Activity No. 6	
SEARCHING TECHNIQUES	
Course Code: CPE010	**Program:** Computer Engineering
Course Title: Data Structures and Algorithms	**Date Performed:** 10 / 15 / 2024
Section: CPE21S4	**Date Submitted:** 10 / 16 / 2024
Name(s): ROALLOS, Jean Gabriel Vincent G.	**Instructor:** Prof. Maria Rizette Sayo
6. Output	

Screenshot



```cpp
#include <iostream>
#include <cstdlib>
#include <time.h>
using namespace std;

const int max_size = 50;

int dataset[max_size];

int main()
{
    srand(time(0));
    for (int i = 0; i < max_size; i++)
    {
        dataset[i] = rand();
    }

    for (int i = 0; i < max_size; i++)
    {
        cout << dataset[i] << " ";
    }
}
```

```
/tmp/mWKJ9JXdLV.o
127504579 374880360 1712474685 264993295 742323001 2116334677 921849495 675712267 1670413514
    217669613 1383888266 456184091 98554153 578655686 677477031 36447526 1742801382 100822058
    1764129406 1349782765 1174224232 219859265 1234404072 1223009828 332638949 1827170613
    1402047470 1351667455 71329657 2038526908 1746474284 198834236 265923621 1311465321 463827531
    1008246622 1280316350 1385677027 1683958889 803246217 1603346640 920363507 1259430308
    1701900793 1499019194 1936907339 1738348320 1094336928 2037729397 1354994078

=== Code Execution Successful ===
```

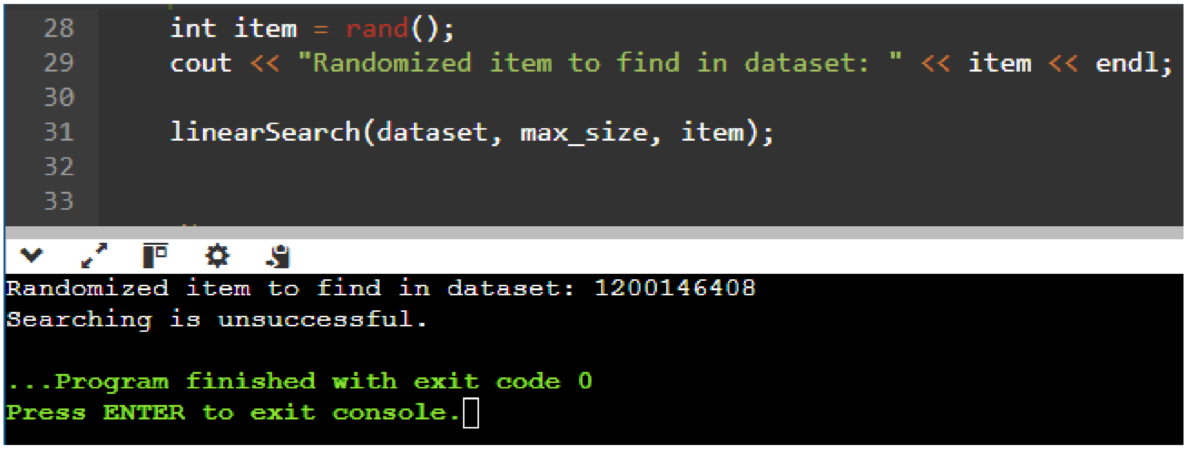
Observations

The code successfully works and the contents of the dataset are randomized every time.

Code

Code is included in screenshot

Output



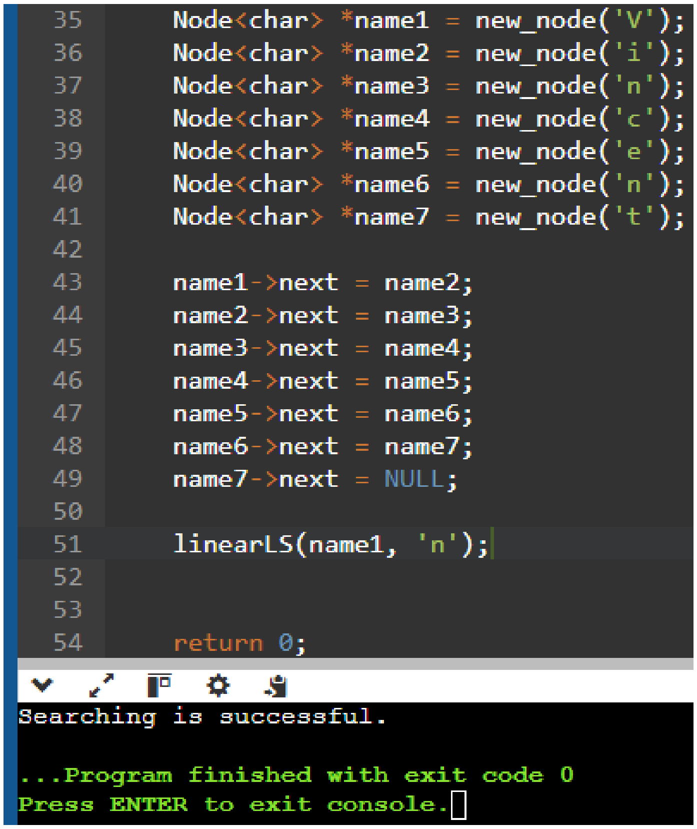
```cpp
    int item = rand();
    cout << "Randomized item to find in dataset: " << item << endl;

    linearSearch(dataset, max_size, item);
```

```
Randomized item to find in dataset: 1200146408
Searching is unsuccessful.

...Program finished with exit code 0
Press ENTER to exit console.
```

Observations

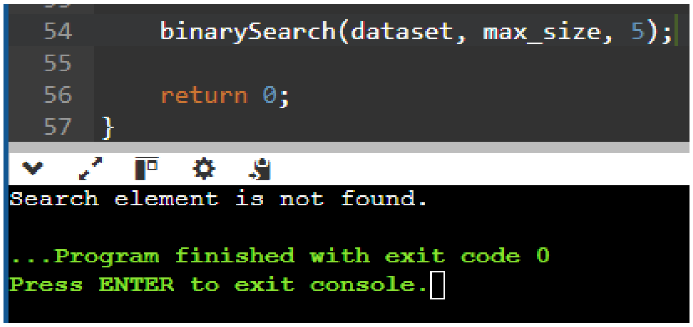
The code works as intended and I assigned a randomized value to the item variable and search it within the dataset array.

Code	*Code is included in screenshot*

Output	
	Node<char> *name1 = new_node('V');
	Node<char> *name2 = new_node('i');
	Node<char> *name3 = new_node('n');
	Node<char> *name4 = new_node('c');
	Node<char> *name5 = new_node('e');
	Node<char> *name6 = new_node('n');
	Node<char> *name7 = new_node('t');
	
	name1->next = name2;
	name2->next = name3;
	name3->next = name4;
	name4->next = name5;
	name5->next = name6;
	name6->next = name7;
	name7->next = NULL;
	
	linearLS(name1, 'n');
	
	return 0;
	Searching is successful.
	
	...Program finished with exit code 0
	Press ENTER to exit console.

Observations	Code worked as intended from initialization of new nodes to searching functions.

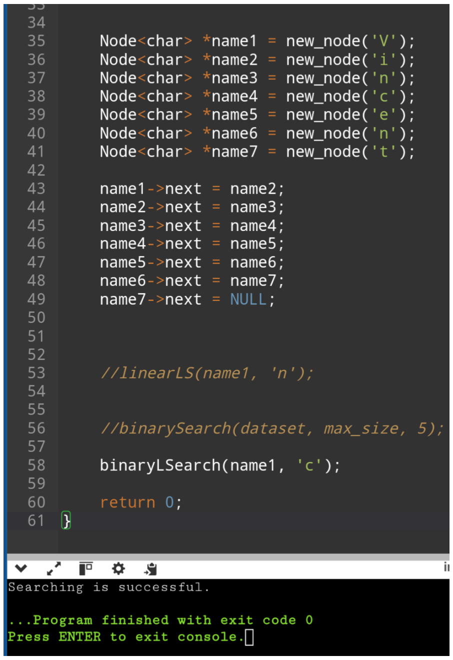
Code	*Code is included in screenshot*

Output	
	binarySearch(dataset, max_size, 5);
	
	return 0;
	}
	Search element is not found.
	
	...Program finished with exit code 0
	Press ENTER to exit console.

Observations	With the use of the first snippet of code, we used the randomzied dataset and searched ifr 5 would be in the array.

Code	*Code is included in screenshot*

Output

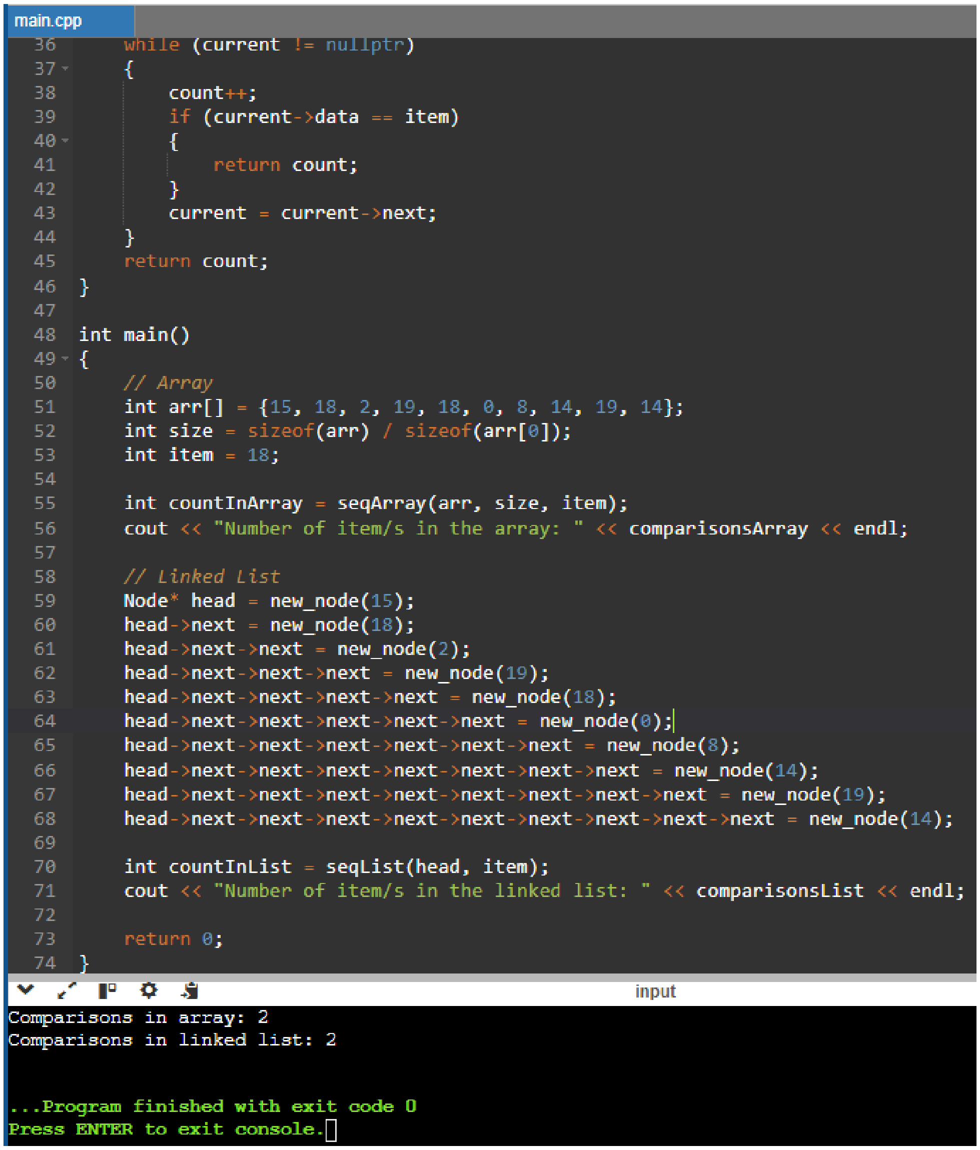


```
    Node<char> *name1 = new_node('V');
    Node<char> *name2 = new_node('i');
    Node<char> *name3 = new_node('n');
    Node<char> *name4 = new_node('c');
    Node<char> *name5 = new_node('e');
    Node<char> *name6 = new_node('n');
    Node<char> *name7 = new_node('t');

    name1->next = name2;
    name2->next = name3;
    name3->next = name4;
    name4->next = name5;
    name5->next = name6;
    name6->next = name7;
    name7->next = NULL;



    //linearLS(name1, 'n');


    //binarySearch(dataset, max_size, 5);

    binaryLSearch(name1, 'c');

    return 0;
}
```

```
Searching is successful.

...Program finished with exit code 0
Press ENTER to exit console.
```

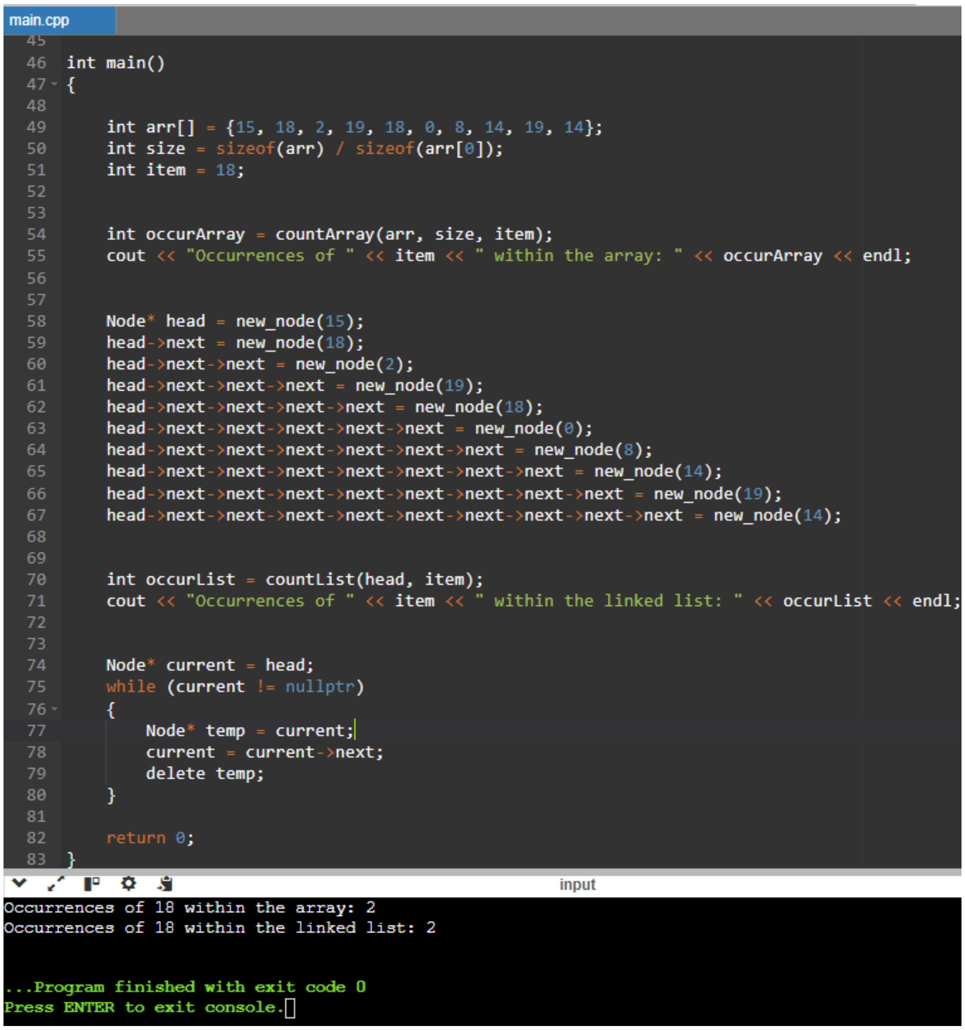
Observations By also using the nodes used and initialized for linear search within the linked list

7. Supplementary Activity



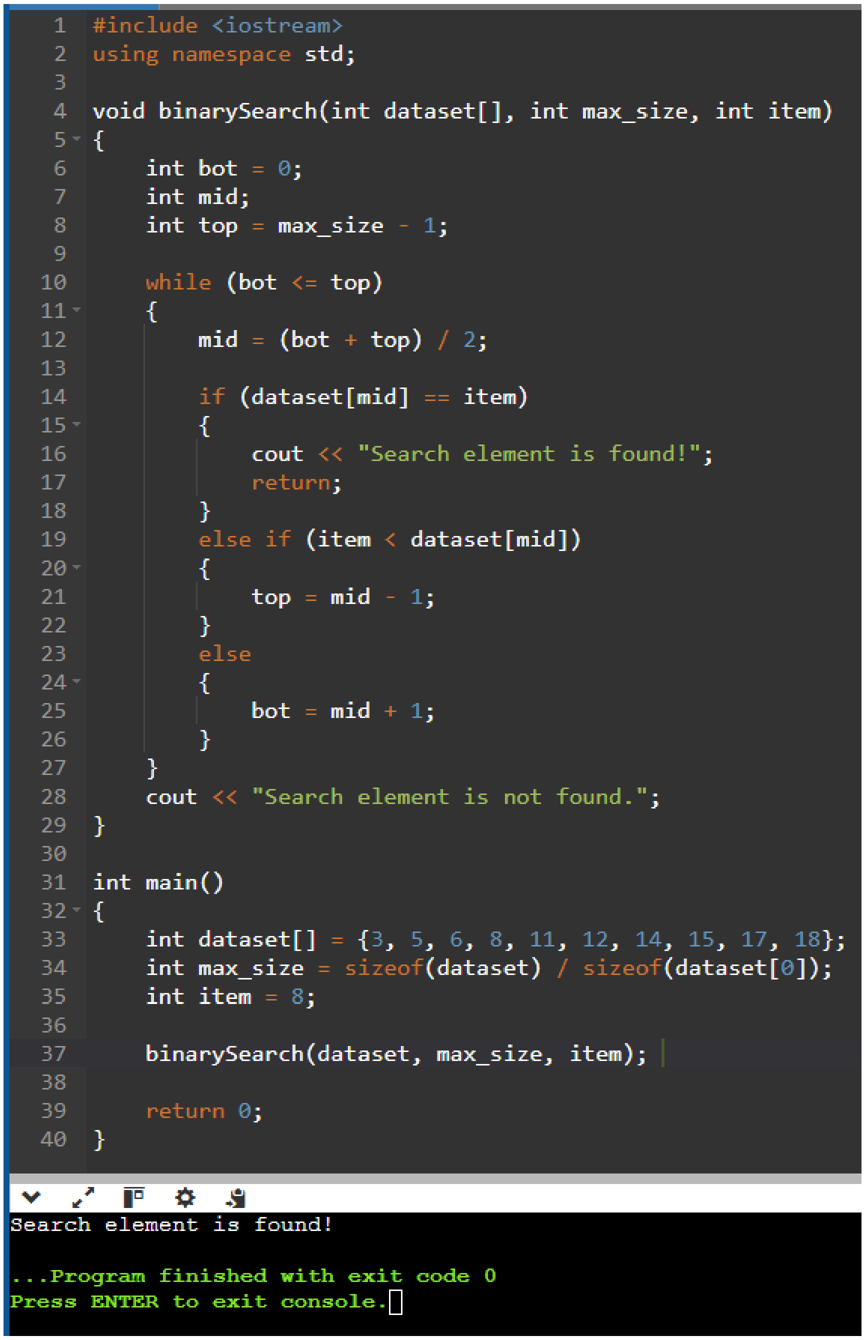
```cpp
    while (current != nullptr)
    {
        count++;
        if (current->data == item)
        {
            return count;
        }
        current = current->next;
    }
    return count;
}

int main()
{
    // Array
    int arr[] = {15, 18, 2, 19, 18, 0, 8, 14, 19, 14};
    int size = sizeof(arr) / sizeof(arr[0]);
    int item = 18;

    int countInArray = seqArray(arr, size, item);
    cout << "Number of item/s in the array: " << comparisonsArray << endl;

    // Linked List
    Node* head = new_node(15);
    head->next = new_node(18);
    head->next->next = new_node(2);
    head->next->next->next = new_node(19);
    head->next->next->next->next = new_node(18);
    head->next->next->next->next->next = new_node(0);
    head->next->next->next->next->next->next = new_node(8);
    head->next->next->next->next->next->next->next = new_node(14);
    head->next->next->next->next->next->next->next->next = new_node(19);
    head->next->next->next->next->next->next->next->next->next = new_node(14);

    int countInList = seqList(head, item);
    cout << "Number of item/s in the linked list: " << comparisonsList << endl;

    return 0;
}
```

```
Comparisons in array: 2
Comparisons in linked list: 2


...Program finished with exit code 0
Press ENTER to exit console.
```



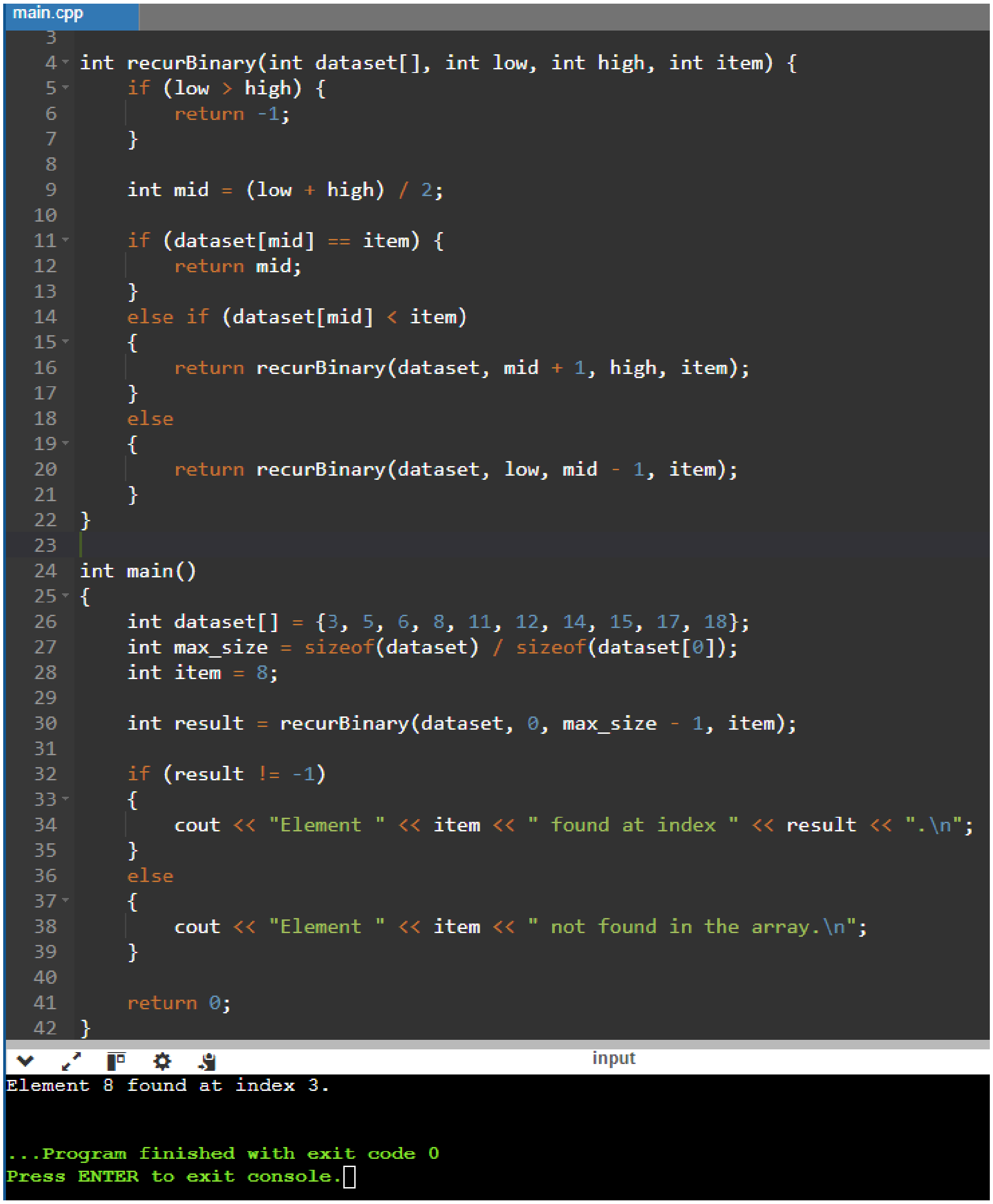
```cpp

int main()
{

    int arr[] = {15, 18, 2, 19, 18, 0, 8, 14, 19, 14};
    int size = sizeof(arr) / sizeof(arr[0]);
    int item = 18;


    int occurArray = countArray(arr, size, item);
    cout << "Occurrences of " << item << " within the array: " << occurArray << endl;


    Node* head = new_node(15);
    head->next = new_node(18);
    head->next->next = new_node(2);
    head->next->next->next = new_node(19);
    head->next->next->next->next = new_node(18);
    head->next->next->next->next->next = new_node(0);
    head->next->next->next->next->next->next = new_node(8);
    head->next->next->next->next->next->next->next = new_node(14);
    head->next->next->next->next->next->next->next->next = new_node(19);
    head->next->next->next->next->next->next->next->next->next = new_node(14);


    int occurList = countList(head, item);
    cout << "Occurrences of " << item << " within the linked list: " << occurList << endl;


    Node* current = head;
    while (current != nullptr)
    {
        Node* temp = current;
        current = current->next;
        delete temp;
    }

    return 0;
}
```

```
Occurrences of 18 within the array: 2
Occurrences of 18 within the linked list: 2


...Program finished with exit code 0
Press ENTER to exit console.
```



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

void binarySearch(int dataset[], int max_size, int item)
{
    int bot = 0;
    int mid;
    int top = max_size - 1;

    while (bot <= top)
    {
        mid = (bot + top) / 2;

        if (dataset[mid] == item)
        {
            cout << "Search element is found!";
            return;
        }
        else if (item < dataset[mid])
        {
            top = mid - 1;
        }
        else
        {
            bot = mid + 1;
        }
    }
    cout << "Search element is not found.";
}

int main()
{
    int dataset[] = {3, 5, 6, 8, 11, 12, 14, 15, 17, 18};
    int max_size = sizeof(dataset) / sizeof(dataset[0]);
    int item = 8;

    binarySearch(dataset, max_size, item);

    return 0;
}
```

```
Search element is found!

...Program finished with exit code 0
Press ENTER to exit console.
```



```cpp

int recurBinary(int dataset[], int low, int high, int item) {
    if (low > high) {
        return -1;
    }

    int mid = (low + high) / 2;

    if (dataset[mid] == item) {
        return mid;
    }
    else if (dataset[mid] < item)
    {
        return recurBinary(dataset, mid + 1, high, item);
    }
    else
    {
        return recurBinary(dataset, low, mid - 1, item);
    }
}

int main()
{
    int dataset[] = {3, 5, 6, 8, 11, 12, 14, 15, 17, 18};
    int max_size = sizeof(dataset) / sizeof(dataset[0]);
    int item = 8;

    int result = recurBinary(dataset, 0, max_size - 1, item);

    if (result != -1)
    {
        cout << "Element " << item << " found at index " << result << ".\n";
    }
    else
    {
        cout << "Element " << item << " not found in the array.\n";
    }

    return 0;
}
```

```
Element 8 found at index 3.


...Program finished with exit code 0
Press ENTER to exit console.
```

8. Conclusion

The process of searching through arrays or linked lists was also done to traversing them, which was also done in the past laboratory activities. The matching process however is a somewhat new process I was not entirely familiar. After this activity, I have learned new things about it and using it to arrays and linked lists. However, I was not able to accomplish supplementary activities due to taking time understanding the concepts of the main laboratory activities.

9. Assessment Rubric