

# LINER SEARCH CODE IMPLEMENTATION

## ARRAY

```
#include <iostream>

using namespace std;

// Function to perform linear search
int linearSearch(int arr[], int size, int key) {
    for (int i = 0; i < size; ++i) {
        if (arr[i] == key) {
            return i; // Return the index of the found element
        }
    }
    return -1; // Return -1 if the element was not found
}

int main() {
    // Define an array and its size
    int arr[] = {10, 20, 30, 40, 50};
    int size = sizeof(arr) / sizeof(arr[0]);

    // Define the key to search for
    int key;
    cout << "Enter the value to search for: ";
    cin >> key;
```

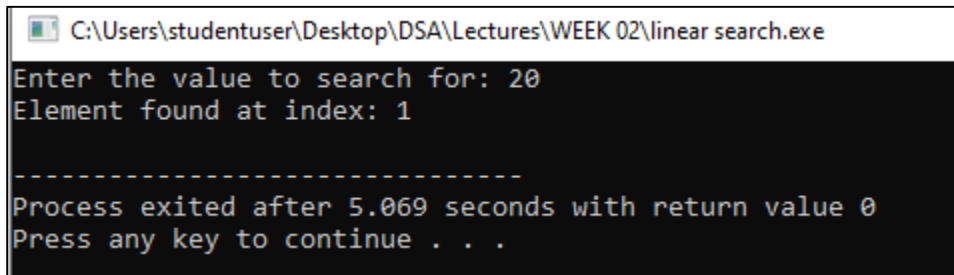
```
// Perform linear search

int index = linearSearch(arr, size, key);


// Output the result
if (index != -1) {
    cout << "Element found at index: " << index << endl;
} else {
    cout << "Element not found in the array." << endl;
}

return 0;
}
```

## OUTPUT



The screenshot shows a Windows command prompt window with the title bar "C:\Users\studentuser\Desktop\DSA\Lectures\WEEK 02\linear search.exe". The prompt displays the following text: "Enter the value to search for: 20", "Element found at index: 1", a dashed line separator, "Process exited after 5.069 seconds with return value 0", and "Press any key to continue . . .".

## LINKED LIST

```
#include <iostream>

using namespace std;

// Node class definition
class Node {
```

```
public:
    int data;
    Node* next;

    Node(int data) : data(data), next(nullptr) {}
};
```

```
// SinglyLinkedList class definition
```

```
class SinglyLinkedList {
private:
    Node* head;

public:
    SinglyLinkedList() : head(nullptr) {}

    // Function to create a new node
    Node* createNode(int data) {
        return new Node(data);
    }
}
```

```
// Function to insert a node at the end
```

```
void insertNodeAtEnd(int data) {
    Node* newNode = createNode(data);
    if (head == nullptr) {
        head = newNode;
    } else {
        Node* temp = head;
```

```

        while (temp->next != nullptr) {
            temp = temp->next;
        }
        temp->next = newNode;
    }
}

// Function to perform linear search
int linearSearch(int key) {
    Node* temp = head;
    int index = 0;
    while (temp != nullptr) {
        if (temp->data == key) {
            return index; // Return the index of the found element
        }
        temp = temp->next;
        ++index;
    }
    return -1; // Return -1 if the element was not found
}

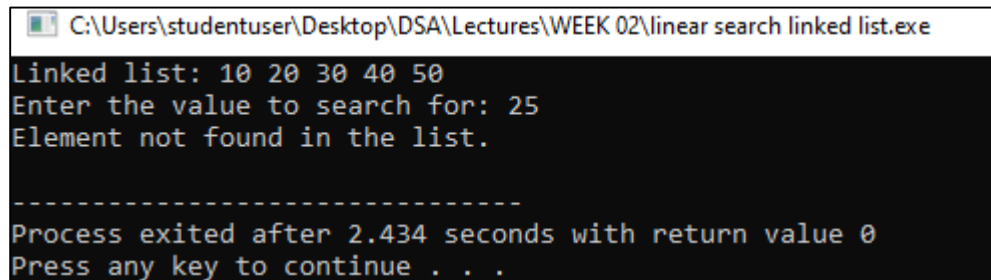
// Function to print the linked list
void printList() {
    Node* temp = head;
    while (temp != nullptr) {
        cout << temp->data << " ";
        temp = temp->next;
    }
}

```

```
    }  
    cout << endl;  
}  
};  
  
// Main function to test the linked list and linear search  
int main() {  
    SinglyLinkedList sll;  
  
    // Insert some elements into the list  
    sll.insertNodeAtEnd(10);  
    sll.insertNodeAtEnd(20);  
    sll.insertNodeAtEnd(30);  
    sll.insertNodeAtEnd(40);  
    sll.insertNodeAtEnd(50);  
  
    cout << "Linked list: ";  
    sll.printList();  
  
    // Perform linear search  
    int key;  
    cout << "Enter the value to search for: ";  
    cin >> key;  
  
    int index = sll.linearSearch(key);  
  
    // Output the result
```

```
if (index != -1) {  
    cout << "Element found at index: " << index << endl;  
} else {  
    cout << "Element not found in the list." << endl;  
}  
  
return 0;  
}
```

## OUTPUT



```
C:\Users\studentuser\Desktop\DSA\Lectures\WEEK 02\linear search linked list.exe  
Linked list: 10 20 30 40 50  
Enter the value to search for: 25  
Element not found in the list.  
-----  
Process exited after 2.434 seconds with return value 0  
Press any key to continue . . .
```