

IMPLEMENTATION OF SINGLY LINKED LIST IN C++

Objective:

The objective of this task is to implement a **Singly Linked List** from scratch using the C++ programming language. The program demonstrates fundamental linked list operations such as insertion, deletion, and searching, along with a driver program to verify the correctness of the implementation.

Description of Singly Linked List:

A **Singly Linked List** is a linear data structure in which each element, called a node, consists of two parts:

- **Data** – stores the value
- **Next Pointer** – stores the address of the next node

Unlike arrays, linked lists do not store elements in contiguous memory locations, allowing dynamic memory allocation and efficient insertion and deletion operations.

Operations Implemented:

The following operations are implemented in the Singly Linked List:

- **Insert** – Adds a new node to the linked list
- **Delete** – Removes a node with a specified value
- **Search** – Searches for an element in the linked list
- **Display** – Displays all elements of the linked list

Time Complexity Analysis:

Operation Time Complexity

Insert O(1)

Delete O(n)

Search O(n)

Display O(n)

Compile and Run Commands:

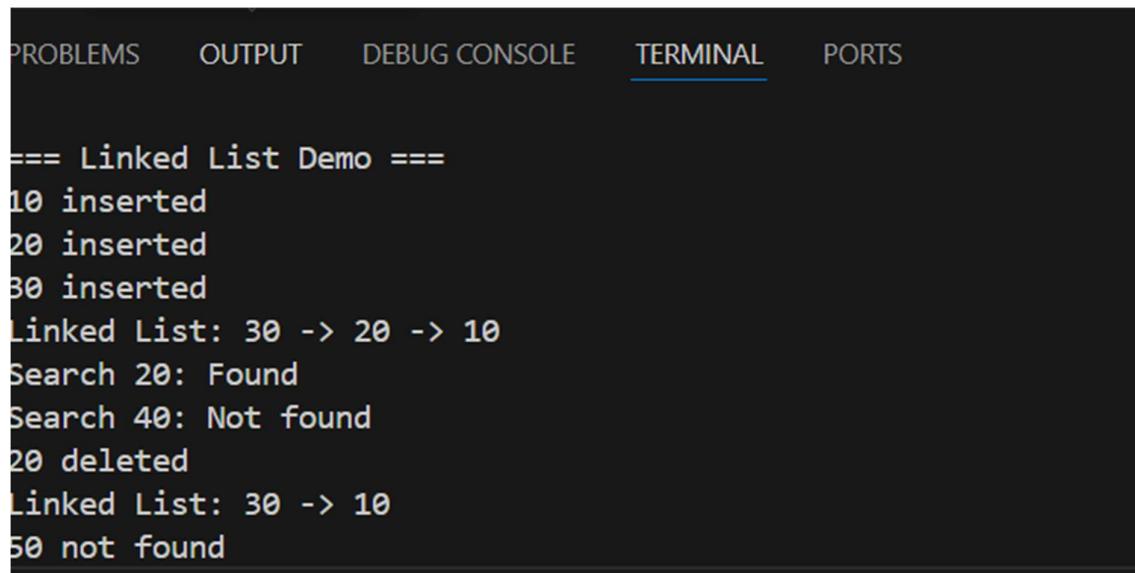
Compile (Using g++):

```
g++ "linked list.cpp" -o linkedlist
```

Run:

```
./linkedlist
```

Sample Input and Output:



The screenshot shows a terminal window with several tabs at the top: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined in blue), and PORTS. The terminal content displays the output of a singly linked list demonstration program. The output is as follows:

```
==== Linked List Demo ====
10 inserted
20 inserted
30 inserted
Linked List: 30 -> 20 -> 10
Search 20: Found
Search 40: Not found
20 deleted
Linked List: 30 -> 10
50 not found
```

Conclusion:

Thus, the Singly Linked List was successfully implemented in C++ using dynamic memory allocation. The program correctly demonstrates insertion, deletion, and search operations along with proper time complexity analysis, fulfilling the requirements of the task.