



**RIPHAH**  
INTERNATIONAL UNIVERSITY

---

**Name = Muhammad Waleed Sattar**

**Sap-id = 55700**

**Section = BS SE 3-2**

**Course = Data Structures and Algorithms**

**Lab Task 5:**

**Q no1:**

**Code & Output:**

FileEditSelectionViewGoRun...←→DSA LAB

Task\_1.cpp ×

LAB5 > Task\_1.cpp > ...

```
1  #include<iostream>
2  using namespace std;
3  //Define the structure for a node
4  class Node
5  {
6      public:
7      int data; //Data to Store
8      Node* next; //Pointer to the next node
9  };
10 //function to insert a node at the beginning
11 void insertAtBeginning(Node** head, int newData){
12     //Allocate memory for a new node
13     Node* newNode = new Node();
14     //Assign data to the new node
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

COMMENTS

Enter number of value you want to add

3

Enter value 1 : 1

Enter value 2 : 2

Enter value 3 : 3

Enter number you want to be deleted: 2

3 ->1 ->NULL

Enter number you want to search: 2

2 is not found in the linked list.

Linked List: 3 ->1 ->NULL

PS D:\DSA\DSA LAB\LAB5>

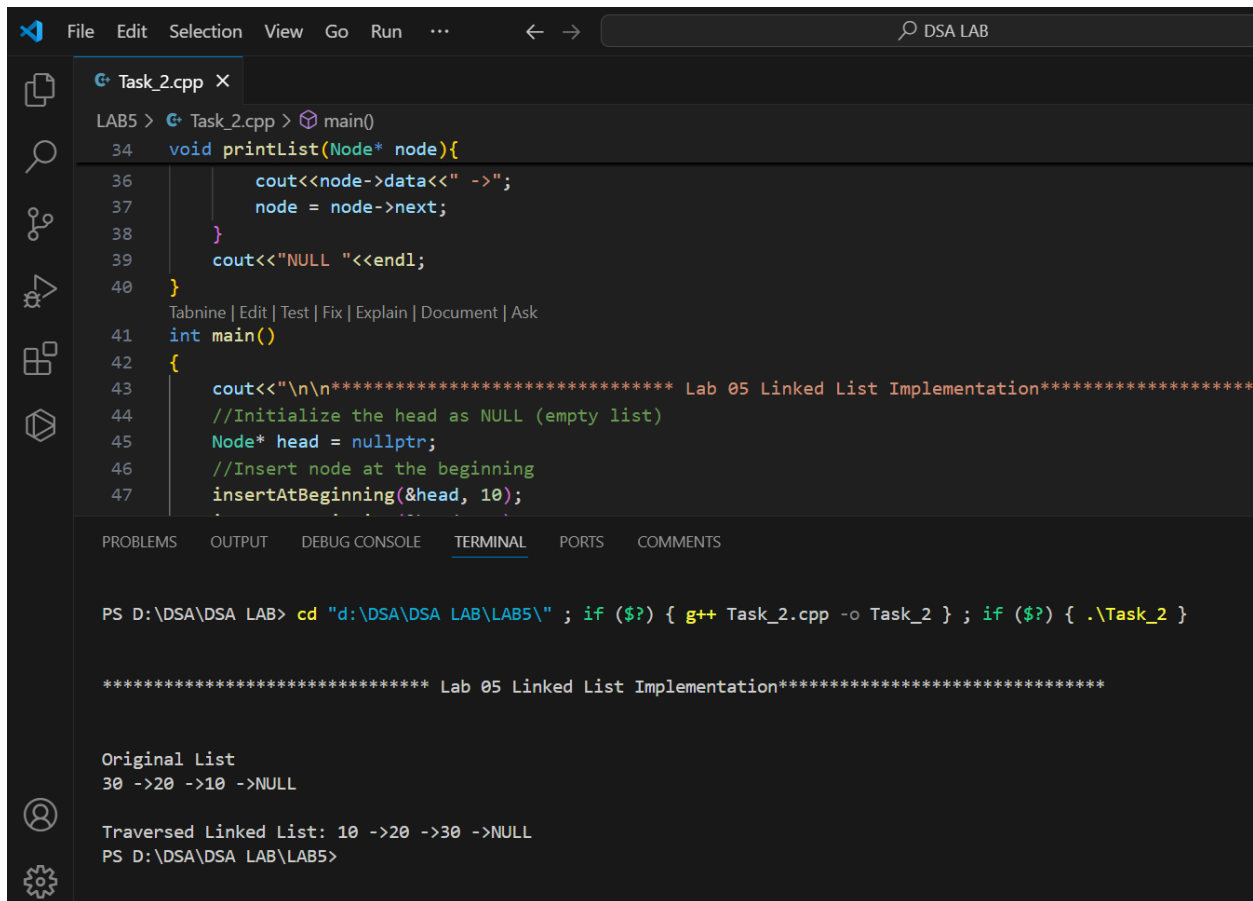
<> 0 0 0 0

31°C  
Haze

Search

Q no 2:

Code & Output:



The screenshot shows a code editor with a file named `Task_2.cpp`. The code implements a linked list with a `printList` function and a `main` function. The `main` function initializes a head pointer to `nullptr` and inserts a node with value 10 at the beginning. The terminal output shows the execution of the program, displaying the original list and the traversed linked list.

```
LAB5 > G Task_2.cpp > main()
34 void printList(Node* node){
36     cout<<node->data<<" ->";
37     node = node->next;
38 }
39 cout<<"NULL " <<endl;
40 }
41 int main()
42 {
43     cout<<"\n\n***** Lab 05 Linked List Implementation*****"
44     //Initialize the head as NULL (empty list)
45     Node* head = nullptr;
46     //Insert node at the beginning
47     insertAtBeginning(&head, 10);
48 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS D:\DSA\DSA LAB> cd "d:\DSA\DSA LAB\LAB5\" ; if ($?) { g++ Task_2.cpp -o Task_2 } ; if ($?) { .\Task_2 }

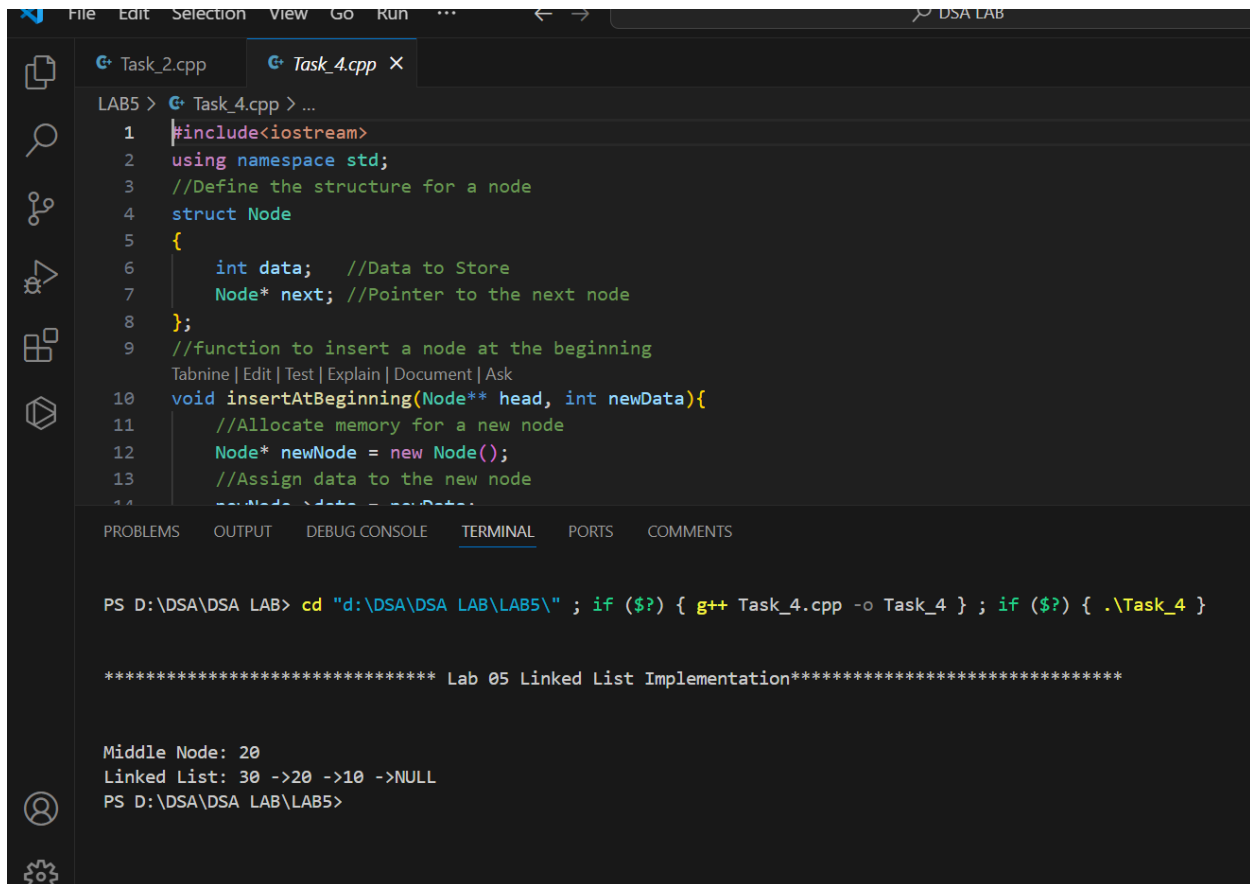
***** Lab 05 Linked List Implementation*****

Original List
30 ->20 ->10 ->NULL

Traversed Linked List: 10 ->20 ->30 ->NULL
PS D:\DSA\DSA LAB\LAB5>
```

Q no 4:

Code & Output:



The screenshot displays a code editor with two tabs: 'Task\_2.cpp' and 'Task\_4.cpp'. The active tab, 'Task\_4.cpp', shows C++ code for a linked list implementation. The code includes headers, namespace declarations, a 'Node' struct, and a function 'insertAtBeginning'. The terminal window at the bottom shows the command to compile and run the program, followed by the output of the program, which includes a separator line, the middle node value, the linked list traversal, and the prompt.

```
LAB5 > G+ Task_4.cpp > ...
1  #include<iostream>
2  using namespace std;
3  //Define the structure for a node
4  struct Node
5  {
6      int data; //Data to Store
7      Node* next; //Pointer to the next node
8  };
9  //function to insert a node at the beginning
10 void insertAtBeginning(Node** head, int newData){
11     //Allocate memory for a new node
12     Node* newNode = new Node();
13     //Assign data to the new node
14     newNode->data = newData;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS D:\DSA\DSA LAB> cd "d:\DSA\DSA LAB\LAB5\" ; if ($?) { g++ Task_4.cpp -o Task_4 } ; if ($?) { .\Task_4 }

***** Lab 05 Linked List Implementation*****

Middle Node: 20
Linked List: 30 ->20 ->10 ->NULL
PS D:\DSA\DSA LAB\LAB5>
```