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

// Node Class
class node
{
public:
   int data;
   node *next;

   // Constructor Function to take input
   node(int d)
   {
       data = d;
       next = NULL;
   }
};

// Function to find length of linkedList
int length(node *head)
{
   int cnt = 0;
   while (head != NULL)
   {
       head = head->next;
       cnt++;
   }
   return cnt;
}

// Insert node at head
void insertAtHead(node *&head, int d)
{
   if (head == NULL)
   {
       head = new node(d);
       return;
   }

   node *n = new node(d);
```

```
    n->next = head;
    head = n;
}

// Insert node at Tail
void insertAtTail(node *&head, int d)
{
    if (head == NULL)
    {
        head = new node(d);
        return;
    }
    int jump = 1;
    node *temp = head;
    while (jump < length(head))
    {
        jump++;
        temp = temp->next;
    }
    node *n = new node(d);
    temp->next = n;
}

// Function to insert at Middle
void insertAtMiddle(node *&head, int data, int p)
{
    if (head == NULL || p == 0 || p == 1)
    {
        insertAtHead(head, data);
        return;
    }

    // jumping to the required node
    int jump = 1;
    node *temp = head;
    while (jump < p - 1)
    {
        jump++;
        temp = temp->next;
    }
```

```
    node *n = new node(data);
    n->next = temp->next;
    temp->next = n;
}

// Delete Head node
void deleteHead(node *&head)
{
    if (head == NULL)
    {
        cout << "Linked List is empty" << endl;
        return;
    }
    node *temp = head->next;
    delete head;
    head = temp;
    cout << "Node deleted Successfully" << endl;
}

// Delete Tail node
void deleteTail(node *&head)
{
    if (head == NULL)
    {
        cout << "Linked List is empty" << endl;
        return;
    }
    int jump = 1;
    node *temp = head;
    node *start = head;
    int p = length(head);
    while (jump < p - 1)
    {
        jump++;
        head = head->next;
    }
    node *last = head;
    head = head->next;
    delete head;
    head = start;
```

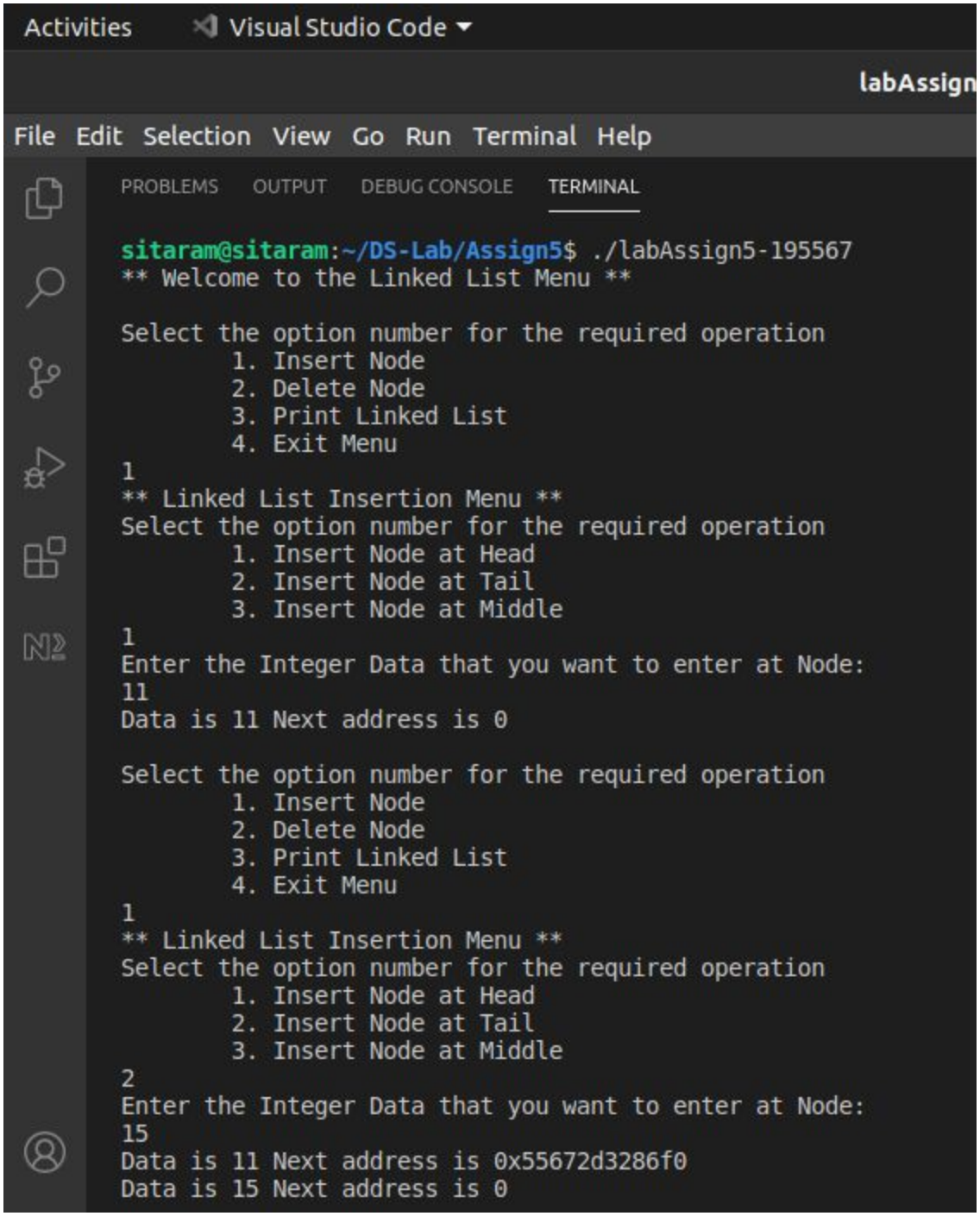
```
    last->next = NULL;
    cout << "Node deleted Successfully" << endl;
}

// Delete node from Middle
void deleteMiddle(node *&head, int p)
{
    if (head == NULL)
    {
        cout << "Linked List is empty" << endl;
        return;
    }
    else if (p > length(head))
    {
        return;
    }
    else if (p == length(head))
    {
        deleteTail(head);
        return;
    }
    else if (p == 1)
    {
        deleteHead(head);
    }
    else
    {
        int jump = 1;
        node *start = head;
        while (jump < p - 1)
        {
            jump++;
            head = head->next;
        }
        node *temp = head;
        head = head->next;
        node *last = head->next;
        delete head;
        head = start;
        temp->next = last;
```

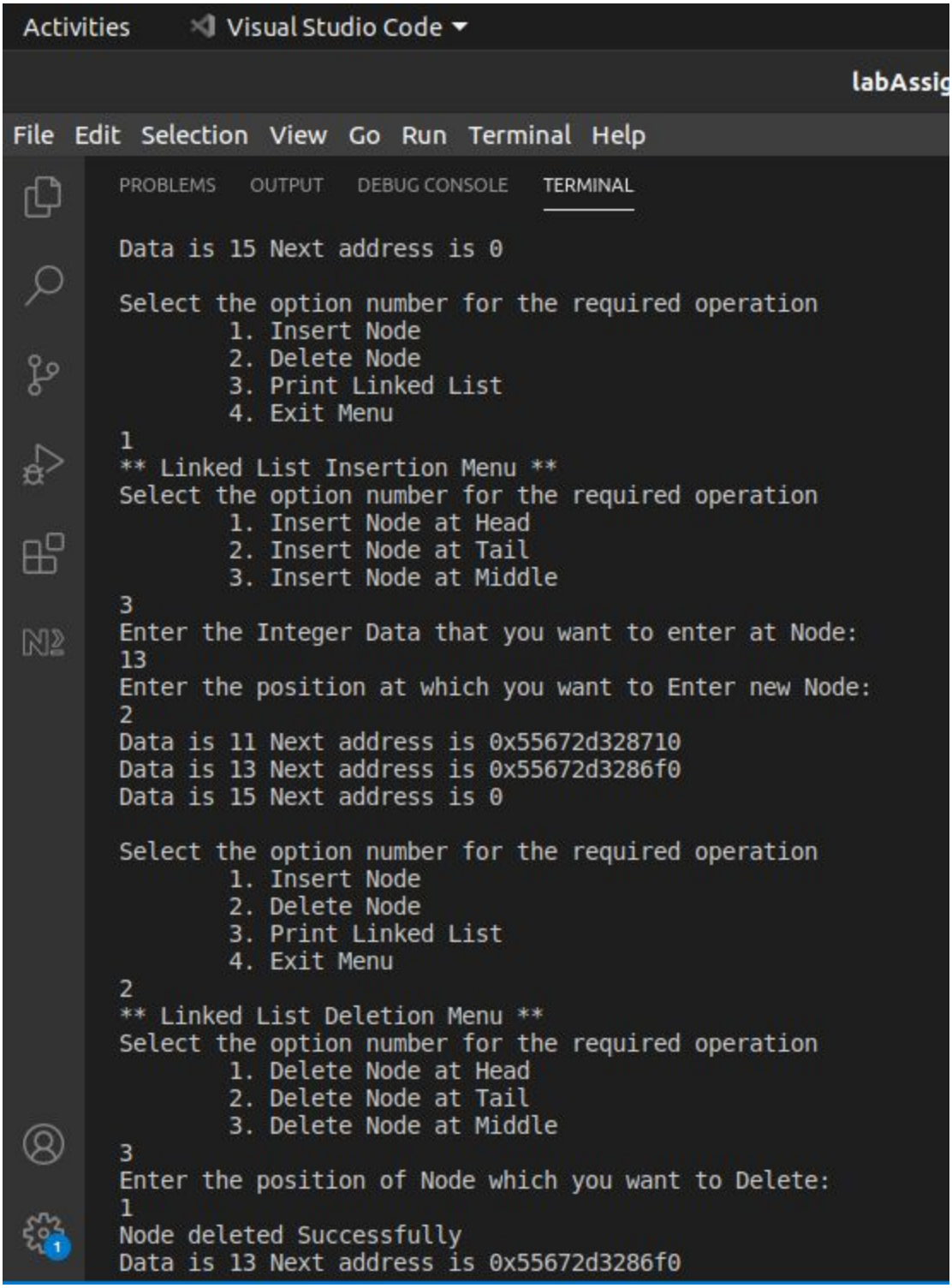
```
        cout << "Node deleted Successfully" << endl;
    }
}

// Function to print Linked List
void print(node *head)
{
    if (head == NULL)
    {
        cout << "Linked List is empty" << endl;
        return;
    }
    while (head != NULL)
    {
        cout << "Data is " << head->data << " Next address is " <<
head->next << endl;
        head = head->next;
    }
}
int main()
{
    node *head = NULL;
    int n, sn, data, pos;
    cout << "** Welcome to the Linked List Menu **" << endl
         << endl;
    cout << "Select the option number for the required operation" << endl;
    cout << "\t1. Insert Node" << endl
         << "\t2. Delete Node" << endl
         << "\t3. Print Linked List" << endl
         << "\t4. Exit Menu" << endl;
    cin >> n;
    while (n != 4)
    {
        switch (n)
        {
        case 1:
            cout << "** Linked List Insertion Menu **" << endl;
            cout << "Select the option number for the required operation"
<< endl;
            cout << "\t1. Insert Node at Head" << endl
```

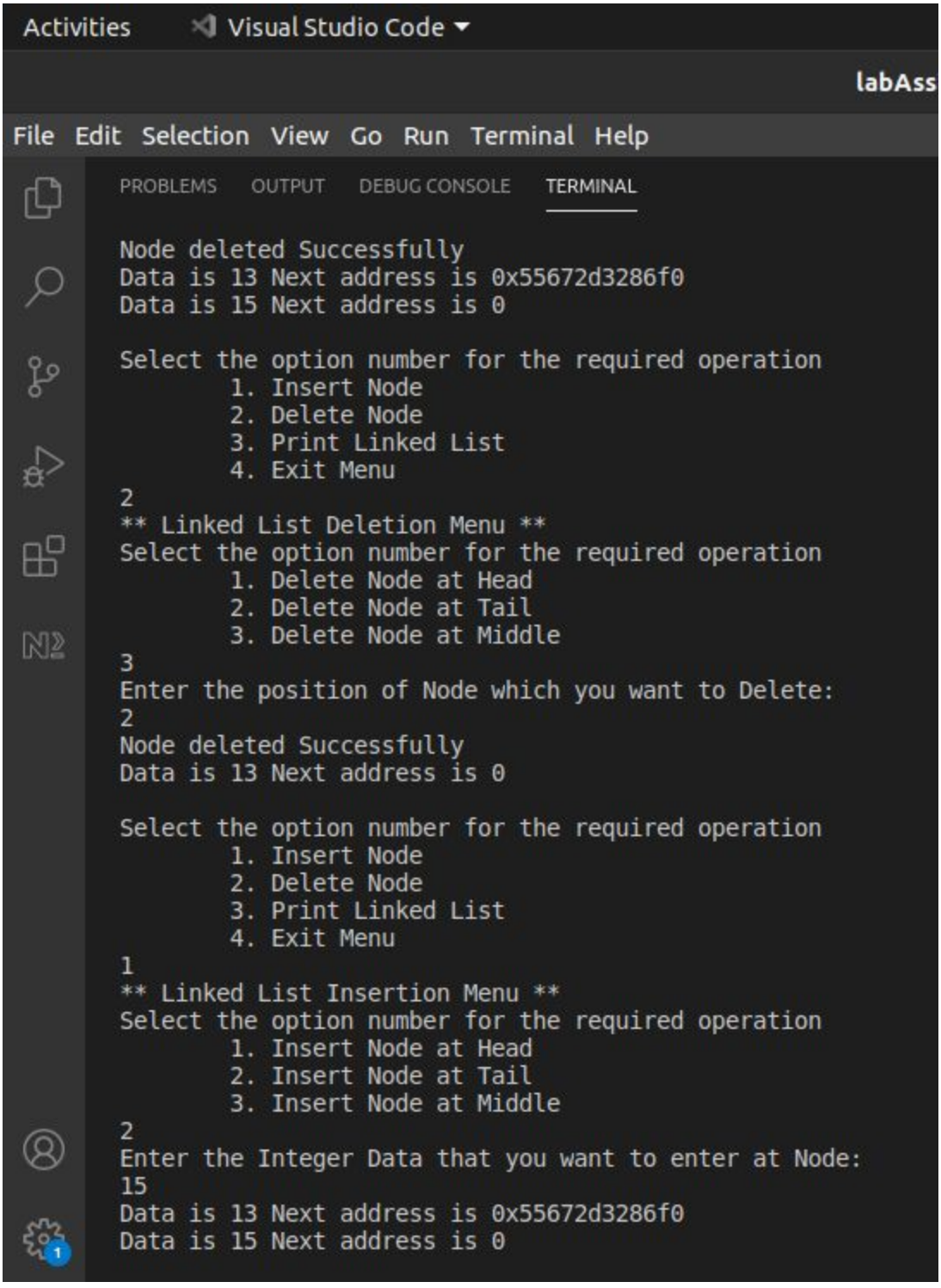
```
                << "\t2. Insert Node at Tail" << endl
                << "\t3. Insert Node at Middle" << endl;
            cin >> sn;
            cout << "Enter the Integer Data that you want to enter at
Node:" << endl;
            cin >> data;
            switch (sn)
            {
            case 1:
                insertAtHead(head, data);
                print(head);
                break;
            case 2:
                insertAtTail(head, data);
                print(head);
                break;
            case 3:
                cout << "Enter the position at which you want to Enter new
Node:" << endl;
                cin >> pos;
                insertAtMiddle(head, data, pos);
                print(head);
                break;
            }

            break;
        case 2:
            cout << "** Linked List Deletion Menu **" << endl;
            cout << "Select the option number for the required operation"
<< endl;
            cout << "\t1. Delete Node at Head" << endl
                << "\t2. Delete Node at Tail" << endl
                << "\t3. Delete Node at Middle" << endl;
            cin >> sn;
            switch (sn)
            {
            case 1:
                deleteHead(head);
                print(head);
                break;
```

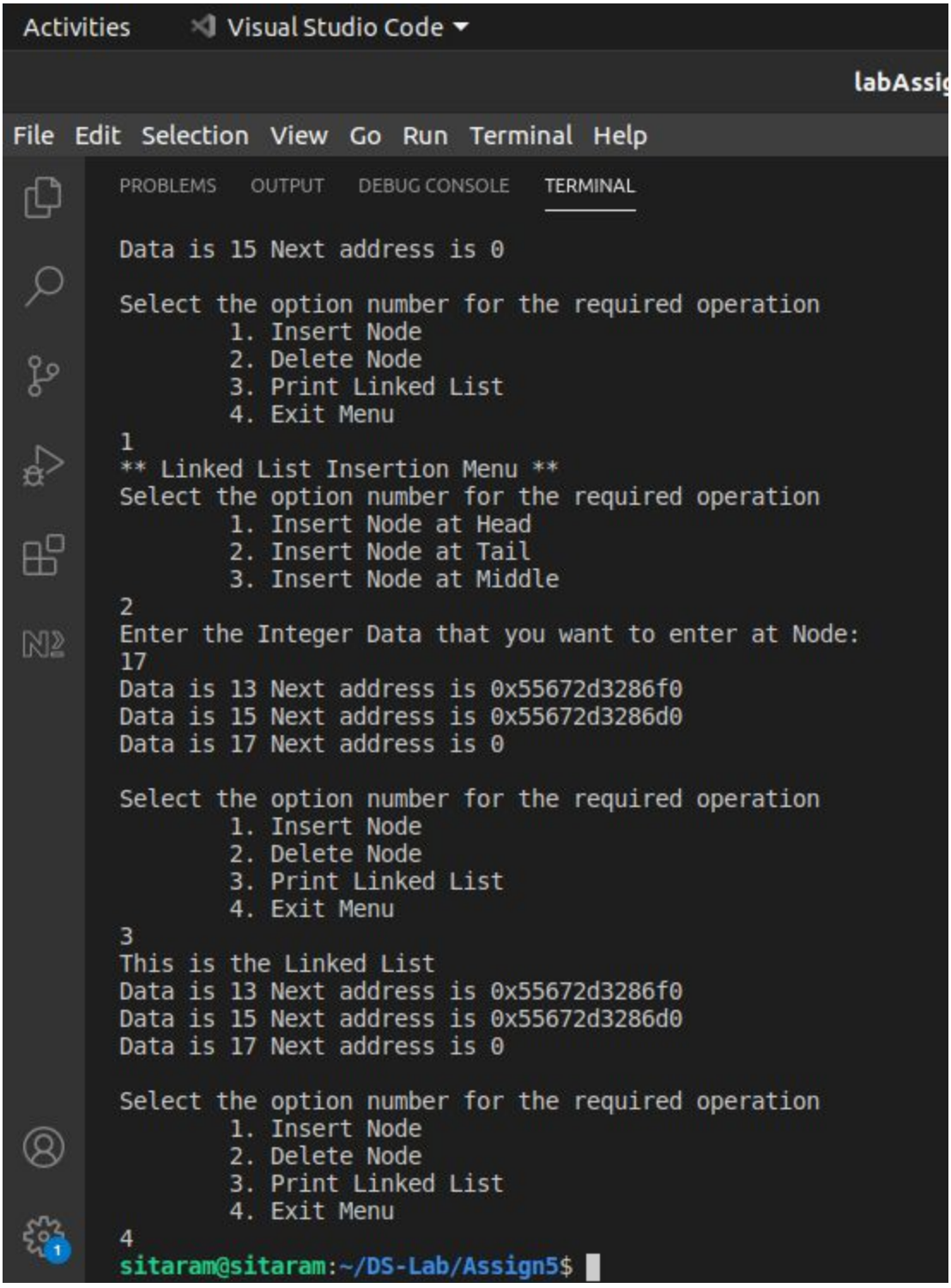
```
            case 2:
                deleteTail(head);
                print(head);
                break;
            case 3:
                cout << "Enter the position of Node which you want to
Delete:" << endl;
                cin >> pos;
                deleteMiddle(head, pos);
                print(head);
                break;
            }
            break;
        case 3:
            cout << "This is the Linked List" << endl;
            print(head);
            break;
        case 4:
            break;
        default:
            cout << "Select the appropriate option Number" << endl;
            break;
        }
        cout << endl;
        cout << "Select the option number for the required operation" <<
endl;
        cout << "\t1. Insert Node" << endl
             << "\t2. Delete Node" << endl
             << "\t3. Print Linked List" << endl
             << "\t4. Exit Menu" << endl;
        cin >> n;
    }

    return 0;
}
```

BELOW IS THE SCREENSHOT OF OUTPUT


Activities
Visual Studio Code
labAssign
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
sitaram@sitaram:~/DS-Lab/Assign5$ ./labAssign5-195567
** Welcome to the Linked List Menu **
Select the option number for the required operation
1. Insert Node
2. Delete Node
3. Print Linked List
4. Exit Menu
1
** Linked List Insertion Menu **
Select the option number for the required operation
1. Insert Node at Head
2. Insert Node at Tail
3. Insert Node at Middle
1
Enter the Integer Data that you want to enter at Node:
11
Data is 11 Next address is 0
Select the option number for the required operation
1. Insert Node
2. Delete Node
3. Print Linked List
4. Exit Menu
1
** Linked List Insertion Menu **
Select the option number for the required operation
1. Insert Node at Head
2. Insert Node at Tail
3. Insert Node at Middle
2
Enter the Integer Data that you want to enter at Node:
15
Data is 11 Next address is 0x55672d3286f0
Data is 15 Next address is 0


Activities
Visual Studio Code
labAssig
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Data is 15 Next address is 0
Select the option number for the required operation
1. Insert Node
2. Delete Node
3. Print Linked List
4. Exit Menu
1
** Linked List Insertion Menu **
Select the option number for the required operation
1. Insert Node at Head
2. Insert Node at Tail
3. Insert Node at Middle
3
Enter the Integer Data that you want to enter at Node:
13
Enter the position at which you want to Enter new Node:
2
Data is 11 Next address is 0x55672d328710
Data is 13 Next address is 0x55672d3286f0
Data is 15 Next address is 0
Select the option number for the required operation
1. Insert Node
2. Delete Node
3. Print Linked List
4. Exit Menu
2
** Linked List Deletion Menu **
Select the option number for the required operation
1. Delete Node at Head
2. Delete Node at Tail
3. Delete Node at Middle
3
Enter the position of Node which you want to Delete:
1
Node deleted Successfully
Data is 13 Next address is 0x55672d3286f0


Activities
Visual Studio Code
labAss
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Node deleted Successfully
Data is 13 Next address is 0x55672d3286f0
Data is 15 Next address is 0
Select the option number for the required operation
1. Insert Node
2. Delete Node
3. Print Linked List
4. Exit Menu
2
** Linked List Deletion Menu **
Select the option number for the required operation
1. Delete Node at Head
2. Delete Node at Tail
3. Delete Node at Middle
3
Enter the position of Node which you want to Delete:
2
Node deleted Successfully
Data is 13 Next address is 0
Select the option number for the required operation
1. Insert Node
2. Delete Node
3. Print Linked List
4. Exit Menu
1
** Linked List Insertion Menu **
Select the option number for the required operation
1. Insert Node at Head
2. Insert Node at Tail
3. Insert Node at Middle
2
Enter the Integer Data that you want to enter at Node:
15
Data is 13 Next address is 0x55672d3286f0
Data is 15 Next address is 0


Activities
Visual Studio Code
labAssig
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Data is 15 Next address is 0
Select the option number for the required operation
1. Insert Node
2. Delete Node
3. Print Linked List
4. Exit Menu
1
** Linked List Insertion Menu **
Select the option number for the required operation
1. Insert Node at Head
2. Insert Node at Tail
3. Insert Node at Middle
2
Enter the Integer Data that you want to enter at Node:
17
Data is 13 Next address is 0x55672d3286f0
Data is 15 Next address is 0x55672d3286d0
Data is 17 Next address is 0
Select the option number for the required operation
1. Insert Node
2. Delete Node
3. Print Linked List
4. Exit Menu
3
This is the Linked List
Data is 13 Next address is 0x55672d3286f0
Data is 15 Next address is 0x55672d3286d0
Data is 17 Next address is 0
Select the option number for the required operation
1. Insert Node
2. Delete Node
3. Print Linked List
4. Exit Menu
4
sitaram@sitaram:~/DS-Lab/Assign5$