Page No.:
Date:

Experiment no. 2

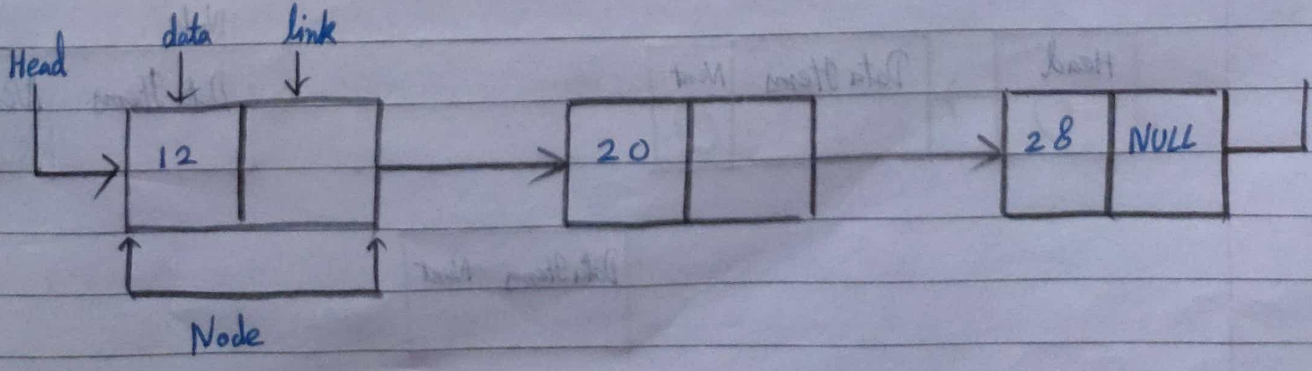
→ Aim - Department of Artificial Intelligence has student's club named .
Students of Second, third and final year of department can be granted membership on request. Similarly one may cancel the membership of club. First node is reserved for president of club & last node is reserved for secretary of club. Write program to maintain club member's information using singly linked list. Store student MIS Registration No. and Name. Write functions to a) Add & delete the members as well as president or even secretary. b) Compute total number of members of club c) Display list in reverse order using recursion d) Display members e) Two linked lists exists for two divisions. Concatenate two list.

→ Theory -

A linked list is a sequence of data structures, which are connected together via links. Linked list is a sequence of links which contains items. Each link contains a connected to another link. Linked list is the second most used data structure after array.
Important terms of Linked list

Link - Each link of a linked list can store a data called an element.
Next - Each link of a linked list contains a link to the next link called Next
Linkedlist - A linked list contain the connection ~~test~~ link to the first link called first

Linked list Representation.



A72 Pratik Jade

Head node is the starting node of the linked list (first node) and it contain the reference to the next node in the list. The head node will have a null reference when the list is empty.

• Types of linked lists

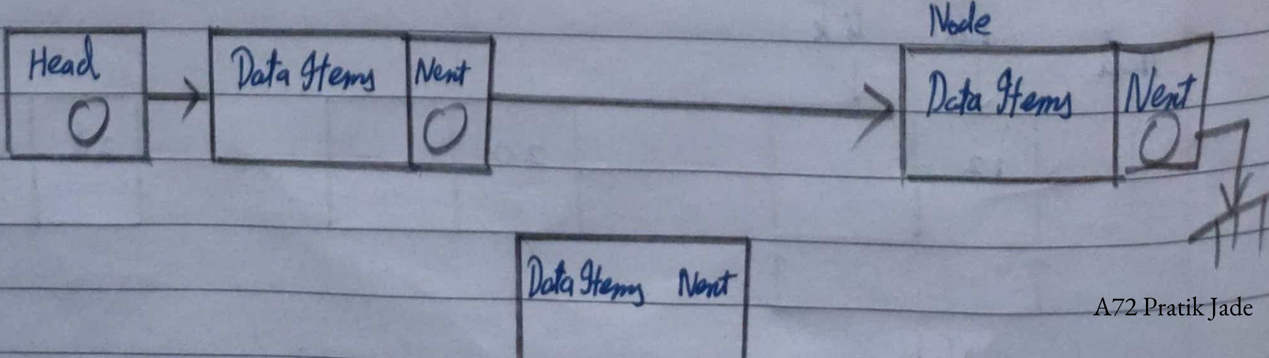
Singly linked list	- Items navigation is forward only.
Doubly linked list	- Items can be navigated forward & backward.
Circular linked list	- Last items contains link of the first element & the first element has a link to the last element as previous

Basic Operations

Insertion	- Adds an element
Deletion	- Deletes an element
Display	- Displays the complete list
Search	- Searches an element
Delete	- Deletes on element

• Insertion Opertion

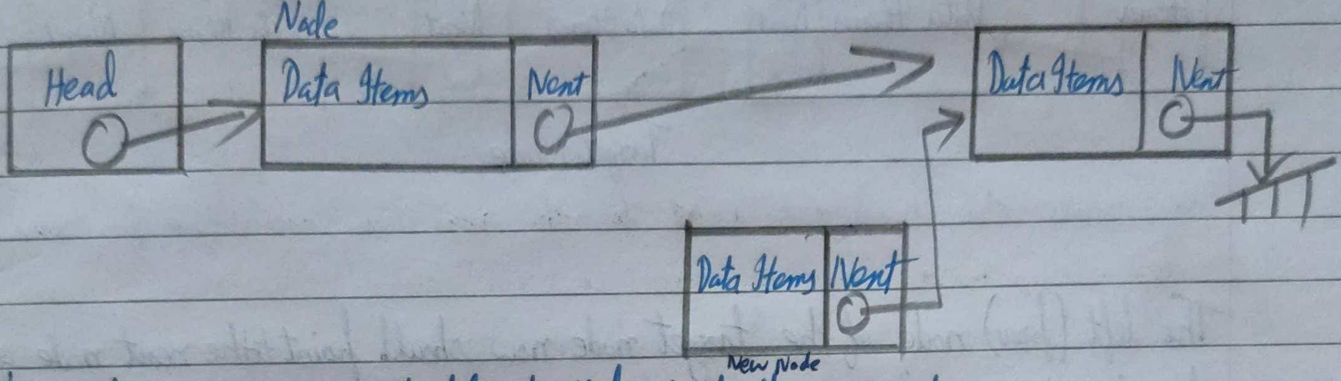
Adding a new node in linked list is a more than one step activity. We shall learn this with diagram here.
First, create a node using the same structure & find the location where it has to be inserted.



We are inserting a node B(new), between A and c. Then point B, next to c-

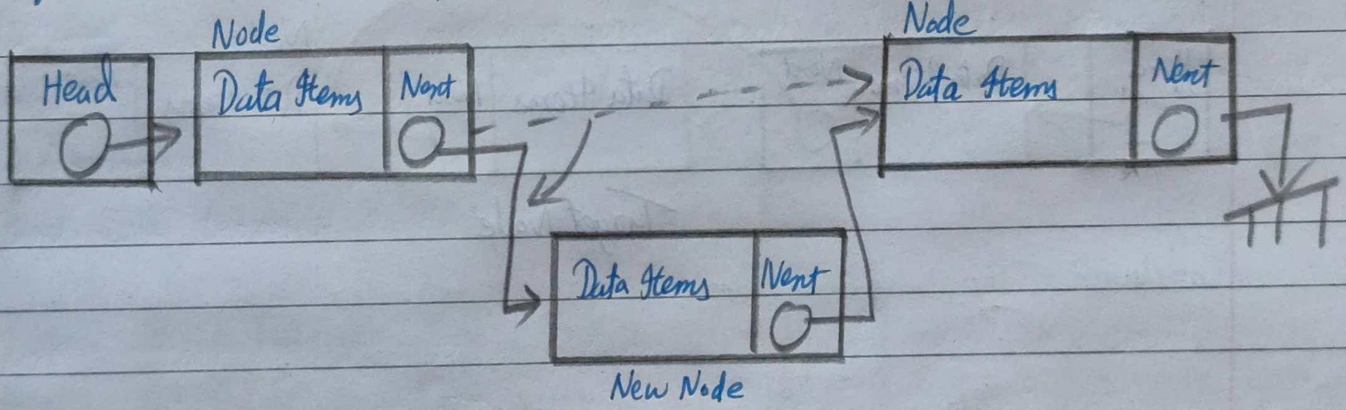
NewNode.next -> RightNode;

It should look like this

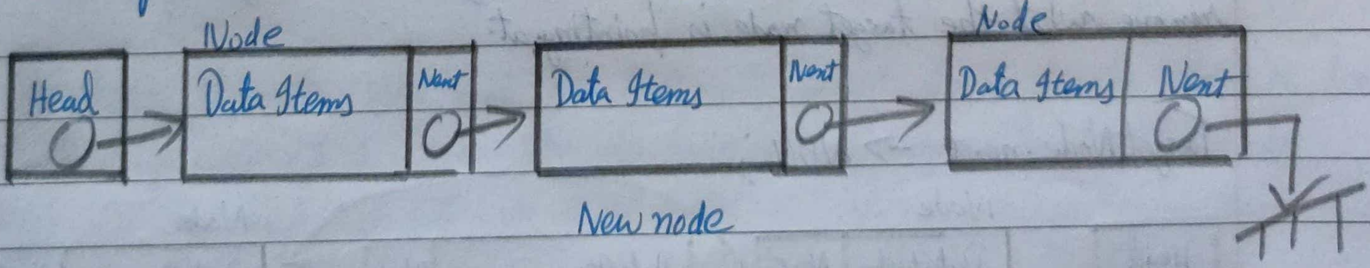


Now, the next node at the left should point to the new node.

LeftNode.next -> NewNode;



This will put the new node in the middle of the two. The new list look like this



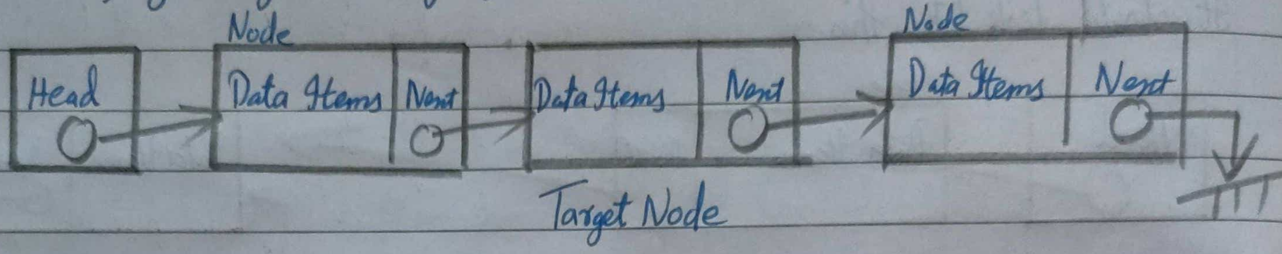
A72 Pratik Jade

Page No.:

Date:

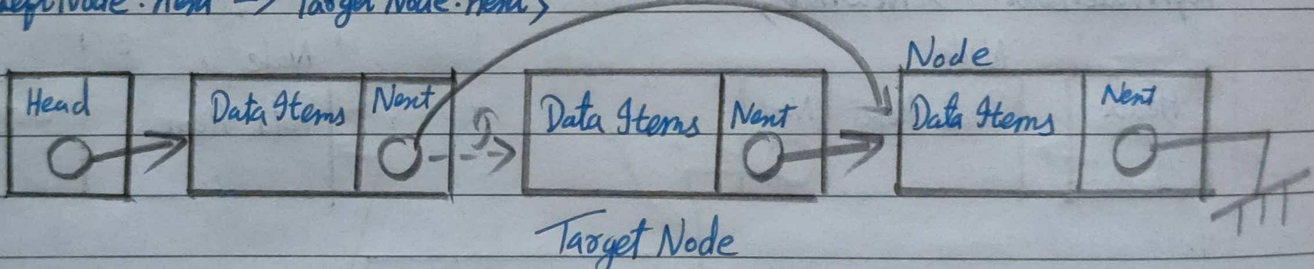
• Deletion Operation.

Deletion is also a more than one step process. First, locate the target node to be removed, by using searching algorithms.



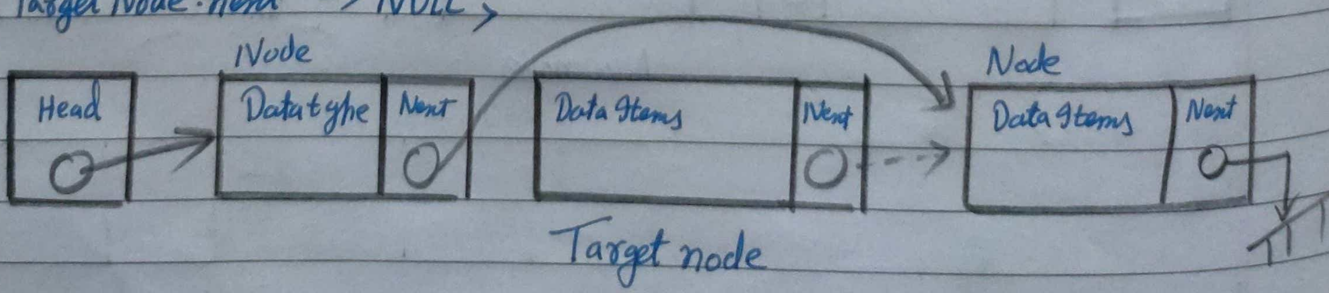
The left (prev) node of the target node now should point to the next node of the target node-

LeftNode.next -> TargetNode.next;



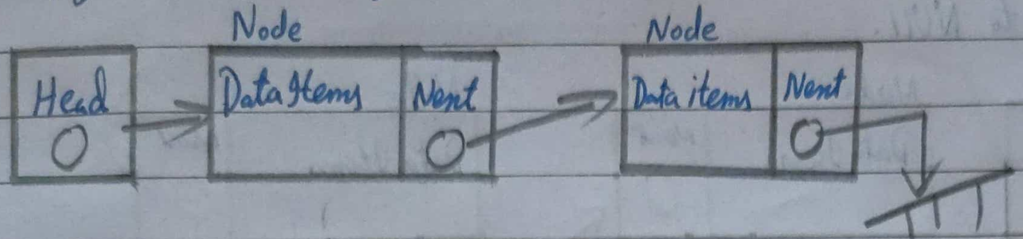
This will remove the link that was pointing to the target node, Now we will remove what the target node is pointing at:

TargetNode.next -> NULL;



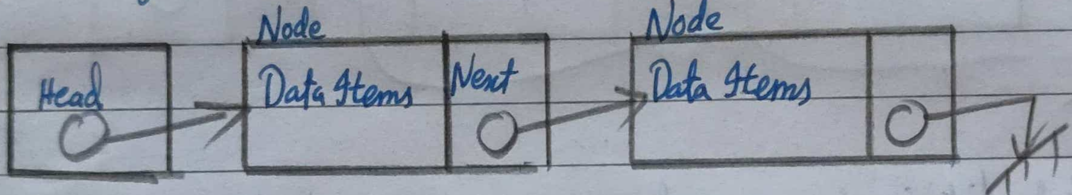
A72 Pratik Jade

We need to use the deleted node. We can keep that in memory otherwise we can simply deallocate memory & wipe off the target node completely.

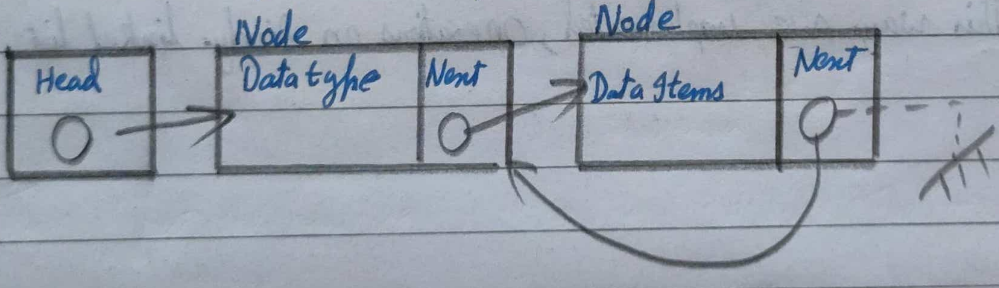


- Reverse Operation.

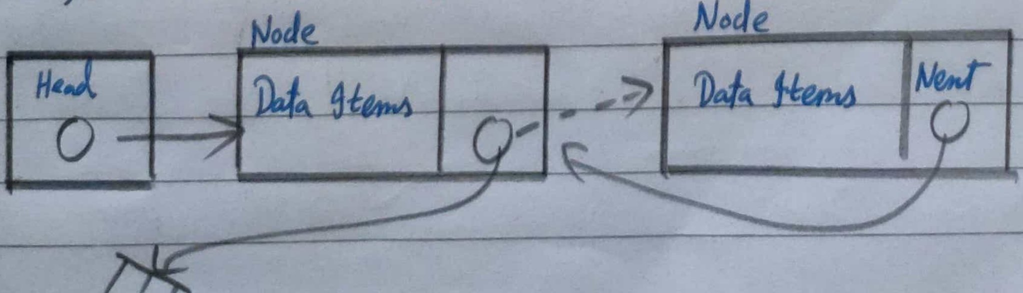
This operation is a thorough one. We need to make the last node to be pointed by the head node & reverse the whole linked list.



First, we traverse to the end of the list. It should be pointed to NULL. Now, we shall make it point to its previous node.



We have to make sure that the last node is not the last node. So, we have some temp node, which looks like the head node pointing to the last node. Now, we shall make all left side nodes point to their previous nodes one by one.

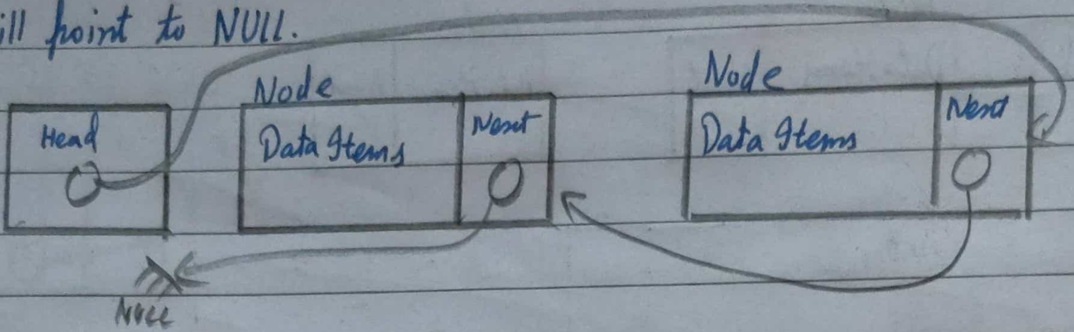


A72 Pratik Jade

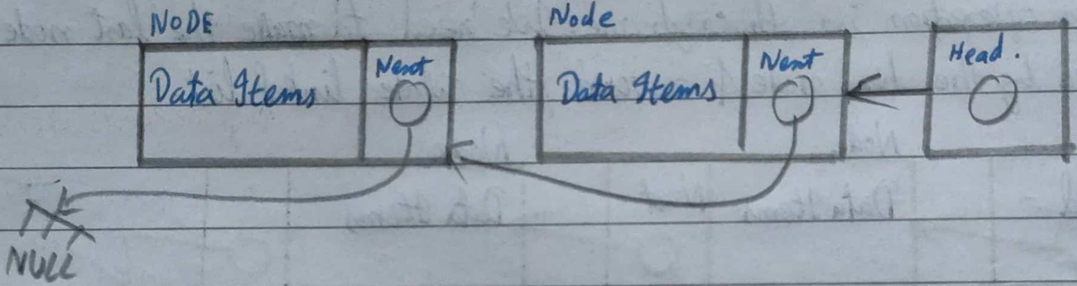
Page No.:

Date:

~~Exp~~ Except the node (first node) pointed by the head node, all nodes should point to their predecessor, making them their new successor. The first node will point to NULL.



We make the head node point to the next new first node by using the temp node.



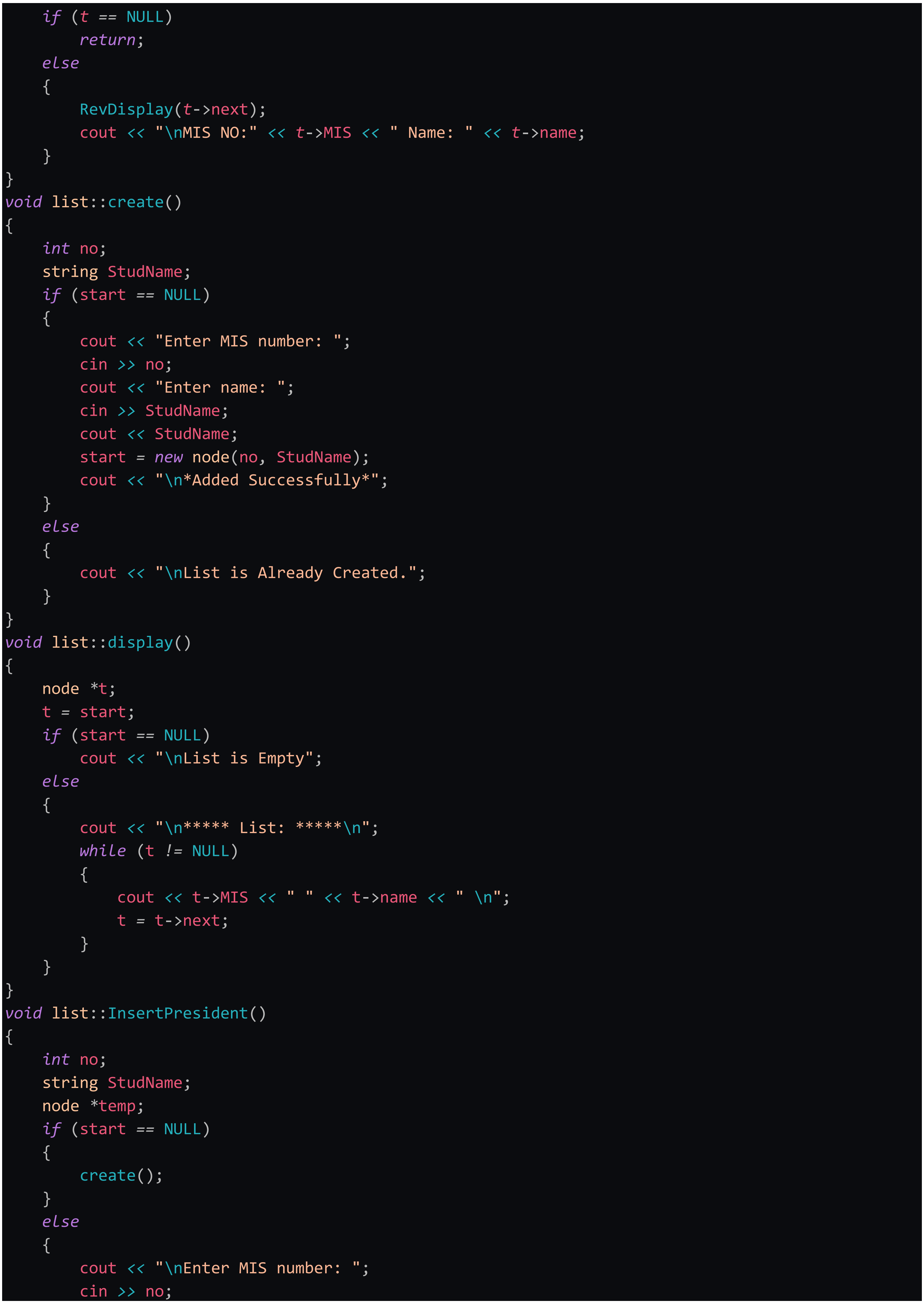
The linked list is now reversed.

Conclusion - This way we implemented, operations on Singly linked list

A72 Pratik Jade

Program:

```cpp
#include <iostream>
#include <string>
using namespace std;
class list;
class node
{
    int MIS;
    string name;
    node *next;

public:
    node(int x, string nem)
    {
        MIS = x;
        next = NULL;
        name = nem;
    }
    friend class list;
};
class list
{
    node *start;

public:
    list()
    {
        start = NULL;
    }
    void create();
    void display();
    void InsertPresident();
    void InsertSecretary();
    void InsertMember();
    void DeletePresident();
    void DeleteMember();
    void DeleteSecretary();
    void SortList();
    void concat(list &q1);
    void RevDisplay(node *t);
    int ContTotal();
    bool DisplayReverse()
    {
        if (start == NULL)
            return false;
        node *temp = start;
        RevDisplay(temp);

        return true;
    }
};
void list::RevDisplay(node *t)
{
```

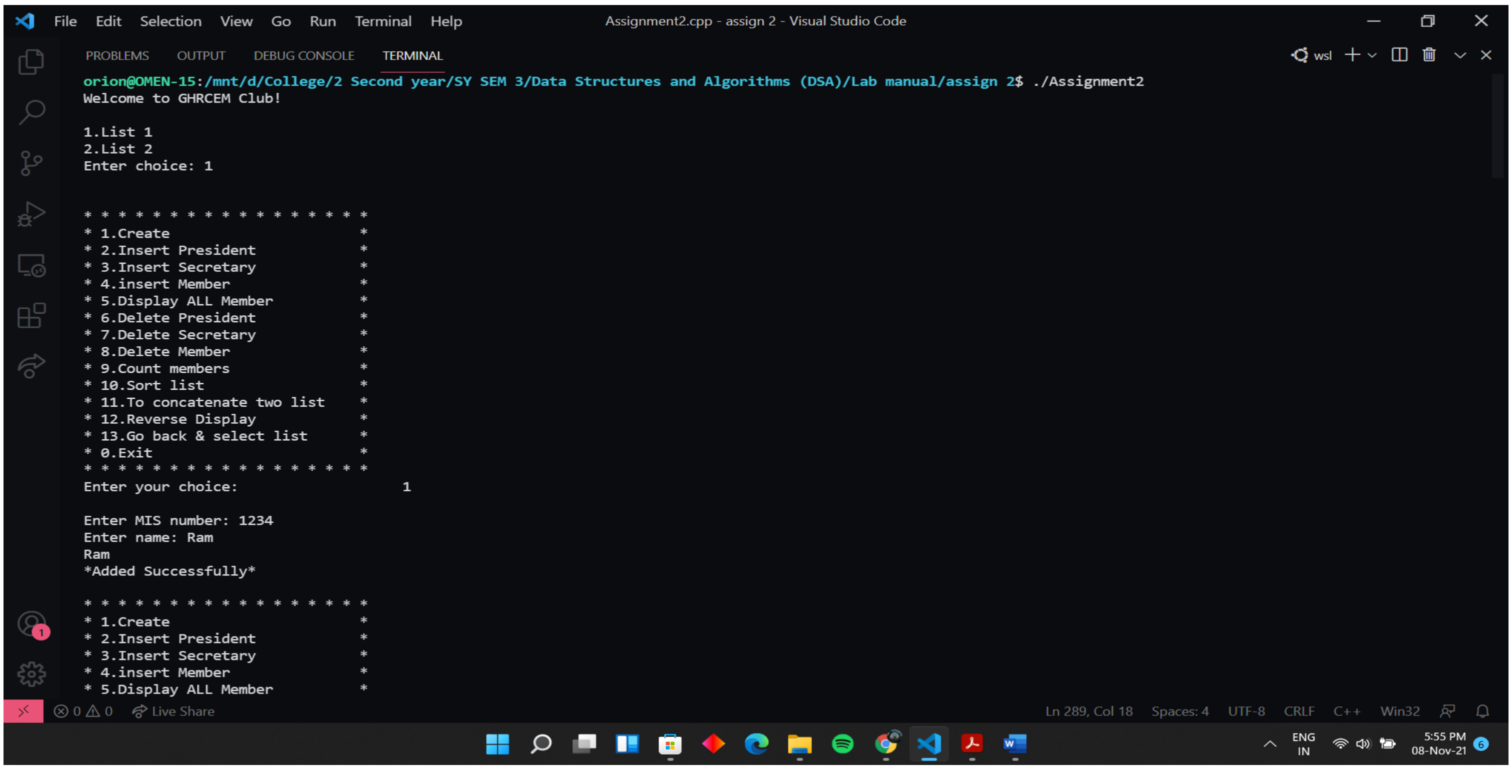


```
    if (t == NULL)
        return;
    else
    {
        RevDisplay(t->next);
        cout << "\nMIS NO:" << t->MIS << " Name: " << t->name;
    }
}
void list::create()
{
    int no;
    string StudName;
    if (start == NULL)
    {
        cout << "Enter MIS number: ";
        cin >> no;
        cout << "Enter name: ";
        cin >> StudName;
        cout << StudName;
        start = new node(no, StudName);
        cout << "\n*Added Successfully*";
    }
    else
    {
        cout << "\nList is Already Created.";
    }
}
void list::display()
{
    node *t;
    t = start;
    if (start == NULL)
        cout << "\nList is Empty";
    else
    {
        cout << "\n***** List: *****\n";
        while (t != NULL)
        {
            cout << t->MIS << " " << t->name << " \n";
            t = t->next;
        }
    }
}
void list::InsertPresident()
{
    int no;
    string StudName;
    node *temp;
    if (start == NULL)
    {
        create();
    }
    else
    {
        cout << "\nEnter MIS number: ";
        cin >> no;
```

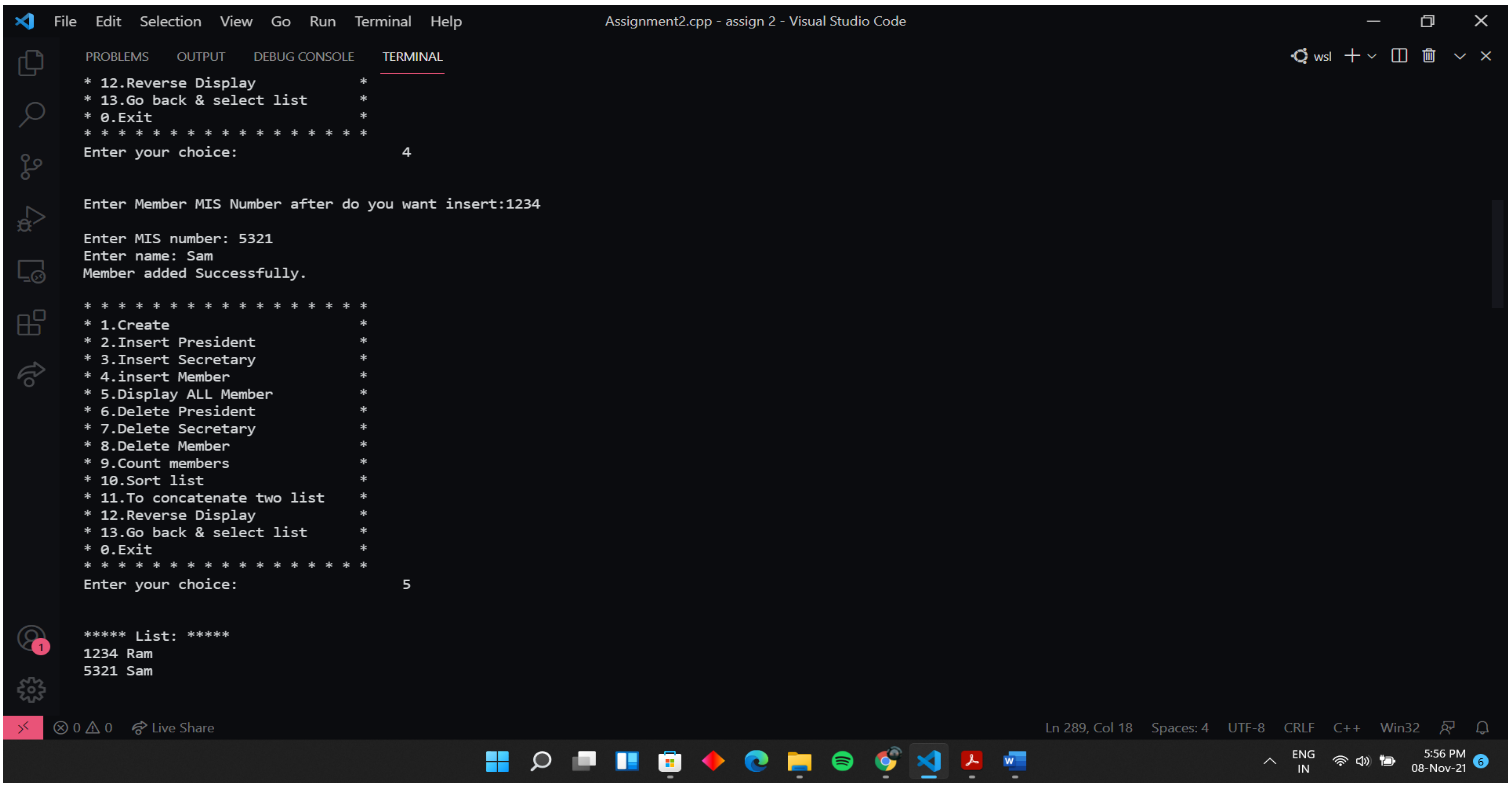
```cpp
        cout << "Enter name: ";
        cin >> StudName;
        temp = new node(no, StudName);
        temp->next = start;
        start = temp;
        //;
        cout << " President " << temp->name << "Inserted Successfully.";
    }
}
void list::InsertSecretary()
{
    int no;
    string StudName;
    node *t;
    if (start == NULL)
        create();
    else
    {
        cout << "\nEnter MIS number: ";
        cin >> no;
        cout << "Enter name: ";
        cin >> StudName;
        t = start;
        while (t->next != NULL)
            t = t->next;
        node *p = new node(no, StudName);
        t->next = p;
    }
    cout << " Secretary Inserted Successfully.";

}
void list::InsertMember()
{
    int prev_no;
    cout << "\nEnter Member MIS Number after do you want insert:";
    cin >> prev_no;
    node *t;
    t = start;
    string StudName;
    int flag = 0, no;
    while (t != NULL)
    {
        if (t->MIS == prev_no)
        {
            flag = 1;
            break;
        }
        t = t->next;
    }
    if (flag == 1)
    {
        node *p;
        cout << "\nEnter MIS number: ";
        cin >> no;
        cout << "Enter name: ";
        cin >> StudName;
        p = new node(no, StudName);
        p->next = t->next;
        t->next = p;
    }
    else
    {
        cout << "\n"
             << prev_no << " Not found.";
    }
    cout << "Member added Successfully.";

}
void list::DeletePresident()
{
    node *t;
    if (start == NULL)
        cout << "\nClub is Empty";
    else
    {
        t = start;
        start = start->next;
        t->next = NULL;
        delete t;
        cout << "\nPresident deleted successfully.";
    }
}
void list::DeleteMember()
{
    int no, flag = 0;
    node *t, *prev;
    if (start == NULL)
        cout << "\nList/Club is empty;";
    else
    {
        cout << "\nEnter member MIS number to be deleted: ";
        cin >> no;
        t = start->next;
        while (t->next != NULL)
        {
            if (t->MIS == no)
            {
                flag = 1;
                break;
            }
            prev = t;
            t = t->next;
        }
        if (flag == 1)
        {
            prev->next = t->next;
            t->next = NULL;
            delete t;
            cout << "\nMember: " << no << " is deleted successfully.";
        }
        else
            cout << "\nMember not Found.";
    }
}
void list::DeleteSecretary()
{
    node *t, *prev;
    t = start;
    if (start == NULL)
        cout << "\nEmpty..";
    else
    {
        while (t->next != NULL)
        {
            prev = t;
            t = t->next;
        }
        prev->next = NULL;
        delete t;
        cout << "\nSecretary Deleted successfully.";
    }
}
int list::ContTotal()
{
    node *t;
    int count = 0;
    t = start;
    if (start == NULL)
    {
        cout << "\nempty.";
        return 0;
    }
    while (t != NULL)
    {
        count++;
        t = t->next;
    }
    return count;
}
void list::SortList()
{
    node *i, *j, *last = NULL;
    int tMIS;
    string tname;
    if (start == NULL)
    {
        cout << "\nempty.";
        return;
    }
    for (i = start; i->next != NULL; i = i->next)
    {
        for (j = start; j->next != last; j = j->next)
        {
            if ((j->MIS) > (j->next->MIS))
            {
                tMIS = j->MIS;
                tname = j->name;
                j->MIS = j->next->MIS;
                j->name = j->next->name;
                j->next->MIS = tMIS;
                j->next->name = tname;
            }
        }
    }
    cout << "\n List is sorted.";
    display();
}
void list::concat(list &q1)
{
    node *t, *p;
    t = q1.start;
    if (t == NULL)
    {
        cout << "\nList 2 is empty";
        return;
    }
    p = start;
    while (p->next != NULL)
    {
        p = p->next;
    }
    p->next = t;
    q1.start = NULL;
    cout << "\nAfter concatenationlist";
    display();
}
int main()
{
    list *l;
    int choice, selectList;
    list l1, l2;
    l = &l1;
X:
    cout << "Welcome to GHRCEM Club!" << endl;
    cout << "\n1.List 1";
    cout << "\n2.List 2";
    cout << "\nEnter choice: ";
    cin >> selectList;
    if (selectList == 1)
    {
        l = &l1;
    }
    else if (selectList == 2)
    {
        l = &l2;
    }
    else
    {
        cout << "\nWrong list Number.";
        goto X;
    }

    do
    {
```

```cpp
        cout << "\n";
        cout << "\n* * * * * * * * * * * * * * * * *";
        cout << "\n* 1.Create                      *";
        cout << "\n* 2.Insert President            *";
        cout << "\n* 3.Insert Secretary            *";
        cout << "\n* 4.insert Member               *";
        cout << "\n* 5.Display ALL Member          *";
        cout << "\n* 6.Delete President            *";
        cout << "\n* 7.Delete Secretary            *";
        cout << "\n* 8.Delete Member               *";
        cout << "\n* 9.Count members               *";
        cout << "\n* 10.Sort list                  *";
        cout << "\n* 11.To concatenate two list    *";
        cout << "\n* 12.Reverse Display            *";
        cout << "\n* 13.Go back & select list      *";
        cout << "\n* 0.Exit                        *";
        cout << "\n* * * * * * * * * * * * * * * * *";
        cout << "\nEnter your choice:\t              ";
        cin >> choice;
        cout << "\n";
        switch (choice)
        {
        case 1:
            l->create();
            break;
        case 2:
            l->InsertPresident();
            break;
        case 3:
            l->InsertSecretary();
            break;
        case 4:
            l->InsertMember();
            break;
        case 5:
            l->display();
            break;
        case 6:
            l->DeletePresident();
            break;
        case 7:
            l->DeleteSecretary();
            break;
        case 8:
            l->DeleteMember();
            break;
        case 9:
            cout << "\nTotal members of Club: " << l->ContTotal();
            break;
        case 10:
            l->SortList();
            break;
        case 11:
            l1.concat(l1);
            break;
        case 12:
            l->DisplayReverse();
            break;
        case 13:
            goto X;
            break;
        deafult:
            cout << "Wrong input try again";
        }
    } while (choice != 0);
    cout << "\nThank you!!\n";
    return 0;
}
```

Output :

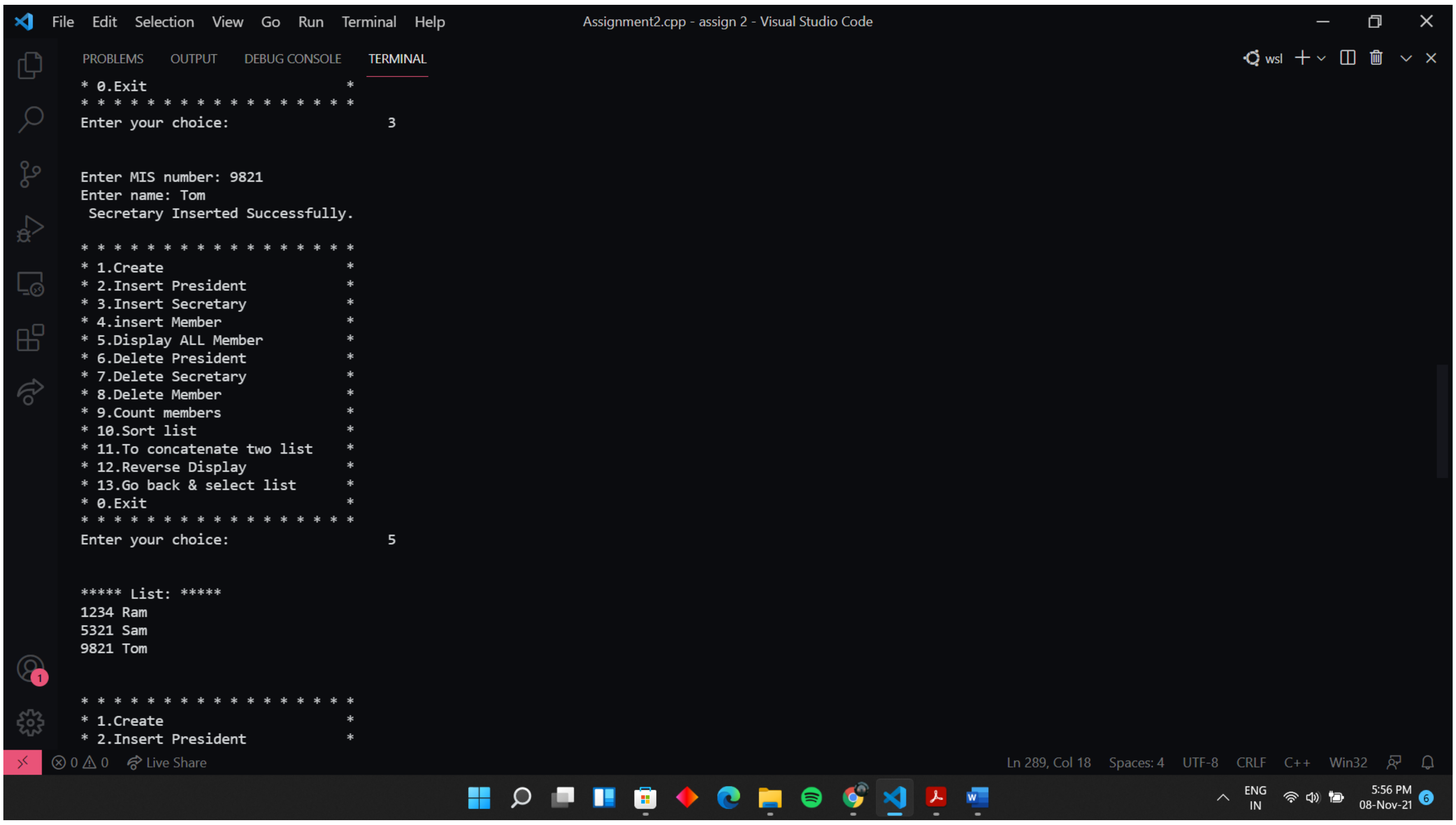


```
orion@OMEN-15:/mnt/d/College/2 Second year/SY SEM 3/Data Structures and Algorithms (DSA)/Lab manual/assign 2$ ./Assignment2
Welcome to GHRCEM Club!

1.List 1
2.List 2
Enter choice: 1


* * * * * * * * * * * * * * * * *
* 1.Create                      *
* 2.Insert President            *
* 3.Insert Secretary            *
* 4.insert Member               *
* 5.Display ALL Member          *
* 6.Delete President            *
* 7.Delete Secretary            *
* 8.Delete Member               *
* 9.Count members               *
* 10.Sort list                  *
* 11.To concatenate two list    *
* 12.Reverse Display            *
* 13.Go back & select list      *
* 0.Exit                        *
* * * * * * * * * * * * * * * * *
Enter your choice:                   1

Enter MIS number: 1234
Enter name: Ram
Ram
*Added Successfully*

* * * * * * * * * * * * * * * * *
* 1.Create                      *
* 2.Insert President            *
* 3.Insert Secretary            *
* 4.insert Member               *
* 5.Display ALL Member          *
```

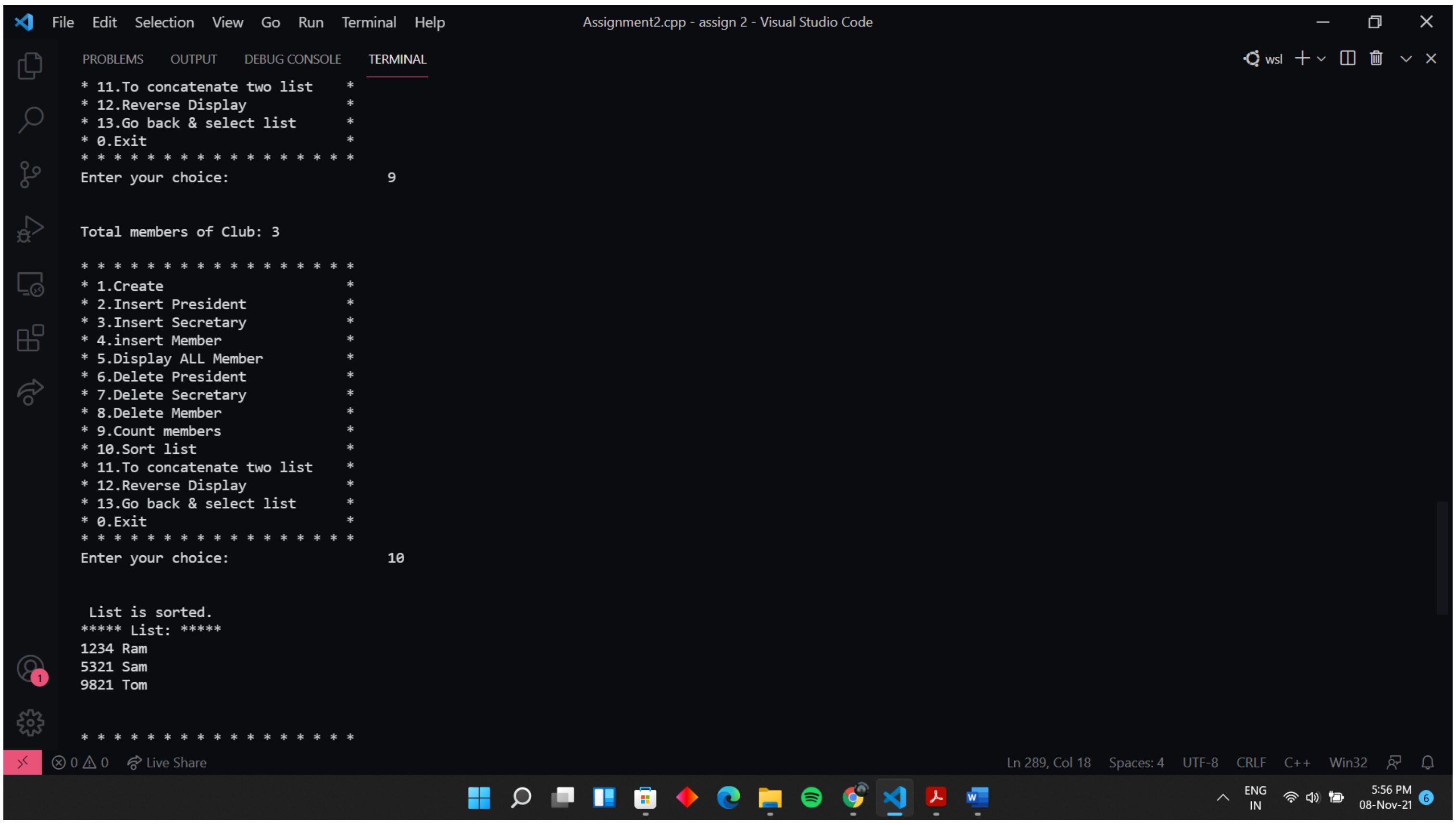


```
* 12.Reverse Display            *
* 13.Go back & select list      *
* 0.Exit                        *
* * * * * * * * * * * * * * * * *
Enter your choice:                   4


Enter Member MIS Number after do you want insert:1234

Enter MIS number: 5321
Enter name: Sam
Member added Successfully.

* * * * * * * * * * * * * * * * *
* 1.Create                      *
* 2.Insert President            *
* 3.Insert Secretary            *
* 4.insert Member               *
* 5.Display ALL Member          *
* 6.Delete President            *
* 7.Delete Secretary            *
* 8.Delete Member               *
* 9.Count members               *
* 10.Sort list                  *
* 11.To concatenate two list    *
* 12.Reverse Display            *
* 13.Go back & select list      *
* 0.Exit                        *
* * * * * * * * * * * * * * * * *
Enter your choice:                   5


***** List: *****
1234 Ram
5321 Sam
```

A72 Pratik Jade

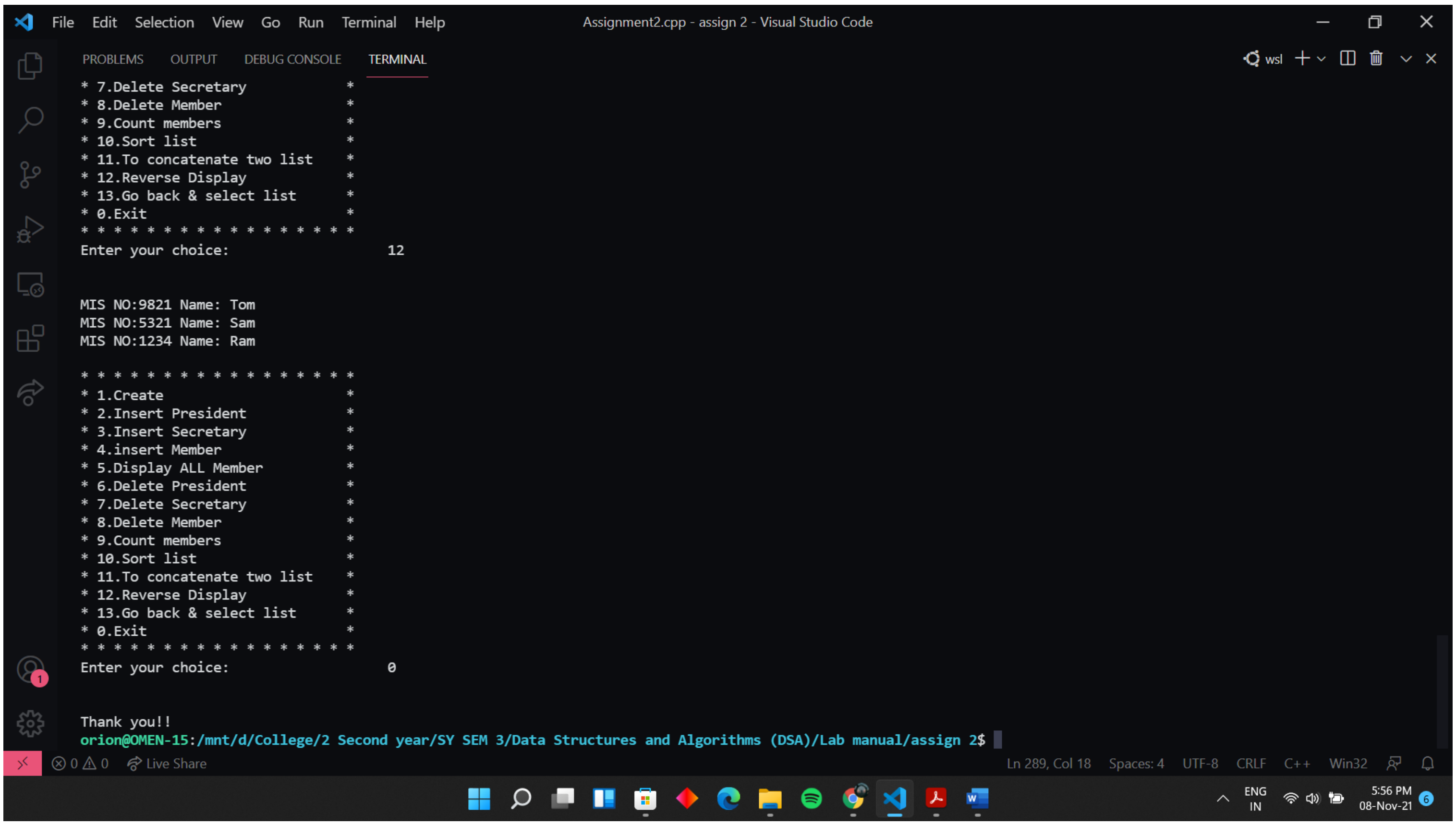

File Edit Selection View Go Run Terminal Help
Assignment2.cpp - assign 2 - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 289, Col 18 Spaces: 4 UTF-8 CRLF C++ Win32
5:56 PM 08-Nov-21

```
* 0.Exit                          *
* * * * * * * * * * * * * * * * * *
Enter your choice:                   3


Enter MIS number: 9821
Enter name: Tom
 Secretary Inserted Successfully.

* * * * * * * * * * * * * * * * * *
* 1.Create                        *
* 2.Insert President              *
* 3.Insert Secretary              *
* 4.insert Member                 *
* 5.Display ALL Member            *
* 6.Delete President              *
* 7.Delete Secretary              *
* 8.Delete Member                 *
* 9.Count members                 *
* 10.Sort list                    *
* 11.To concatenate two list      *
* 12.Reverse Display              *
* 13.Go back & select list        *
* 0.Exit                          *
* * * * * * * * * * * * * * * * * *
Enter your choice:                   5


***** List: *****
1234 Ram
5321 Sam
9821 Tom


* * * * * * * * * * * * * * * * * *
* 1.Create                        *
* 2.Insert President              *
```


File Edit Selection View Go Run Terminal Help
Assignment2.cpp - assign 2 - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 289, Col 18 Spaces: 4 UTF-8 CRLF C++ Win32
5:56 PM 08-Nov-21

```
* 11.To concatenate two list      *
* 12.Reverse Display              *
* 13.Go back & select list        *
* 0.Exit                          *
* * * * * * * * * * * * * * * * * *
Enter your choice:                   9


Total members of Club: 3

* * * * * * * * * * * * * * * * * *
* 1.Create                        *
* 2.Insert President              *
* 3.Insert Secretary              *
* 4.insert Member                 *
* 5.Display ALL Member            *
* 6.Delete President              *
* 7.Delete Secretary              *
* 8.Delete Member                 *
* 9.Count members                 *
* 10.Sort list                    *
* 11.To concatenate two list      *
* 12.Reverse Display              *
* 13.Go back & select list        *
* 0.Exit                          *
* * * * * * * * * * * * * * * * * *
Enter your choice:                   10


 List is sorted.
***** List: *****
1234 Ram
5321 Sam
9821 Tom


* * * * * * * * * * * * * * * * * *
```

A72 Pratik Jade



```
* 7.Delete Secretary          *
* 8.Delete Member             *
* 9.Count members             *
* 10.Sort list                *
* 11.To concatenate two list  *
* 12.Reverse Display          *
* 13.Go back & select list    *
* 0.Exit                      *
* * * * * * * * * * * * * * * * *
Enter your choice:                  12


MIS NO:9821 Name: Tom
MIS NO:5321 Name: Sam
MIS NO:1234 Name: Ram

* * * * * * * * * * * * * * * * *
* 1.Create                    *
* 2.Insert President          *
* 3.Insert Secretary          *
* 4.insert Member             *
* 5.Display ALL Member        *
* 6.Delete President          *
* 7.Delete Secretary          *
* 8.Delete Member             *
* 9.Count members             *
* 10.Sort list                *
* 11.To concatenate two list  *
* 12.Reverse Display          *
* 13.Go back & select list    *
* 0.Exit                      *
* * * * * * * * * * * * * * * * *
Enter your choice:                  0


Thank you!!
orion@OMEN-15:/mnt/d/College/2 Second year/SY SEM 3/Data Structures and Algorithms (DSA)/Lab manual/assign 2$
```

A72 Pratik Jade