G. H. Raisoni College Of Engineering And Management, Wagholi Pune

2021- 2022

Assignment no :- 2

| | | | |
|---|---|---|---|
| Department | CE [SUMMER 2022 (Online)] | | |
| Term / Section | III/B | Date Of **submission** | 12-10-2021 |
| Subject Name /Code | Data Structures and Algorithms/ UCSL201/UCSP201 | | |
| Roll No. | SCOB77 | Name | Pratham Rajkumar pitty |
| Registration Number | 2020AC0E1100107 | | |

# Assignment No 2

**Aim:→** Department of Computer Engineering has Student club named 'COMET'. 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 and last node is reserved for secretary of club. Write program to maintain club member's information using single linked list. Store student MIS Registration No. and name, write Functions to (a) Add and delete the members as well as president or even secretary
(b) compute total no. of members of club
(c) Display members
(d) Display List in Reverse order using recursion
(e) Two linked lists exist for 2 division concatenate two lists.

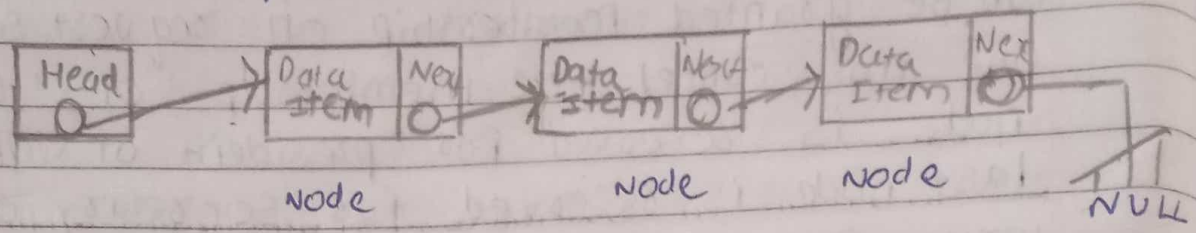
**Theory:→** A link list is a sequence of data structure which is connected through links. It is a sequence of links which contains items. Each link contains connection to another link. Linklist is the second most used data structure after array.

- **Important terms to understand linklist:**

- **Link →** Each link of a linked list can store data called an element.
- **Next →** Each link contains a link to the next link called Next/Node.
- **Linked List →** A Linked List contains the connection link to the first link called first/Head

## Linked List Representation

Linked List can be visualized as a chain of nodes where every node points to the next node.




Where,

- Linked List contains a link element called First.
- Each link carries a data field (s) and a linked field called Next.
- Each link is linked with its next link using its next link
- Last link carries a link as null to mark the end of the list.

## Types of Linked List

- Simple Linked list → Item navigation in forword only
- Doubly Linked List → Items can be navigated Forword & backward
- Circular linked List → Last item contains link of the First element as next and the First element has a link to the last element as previous.

## Basic operation

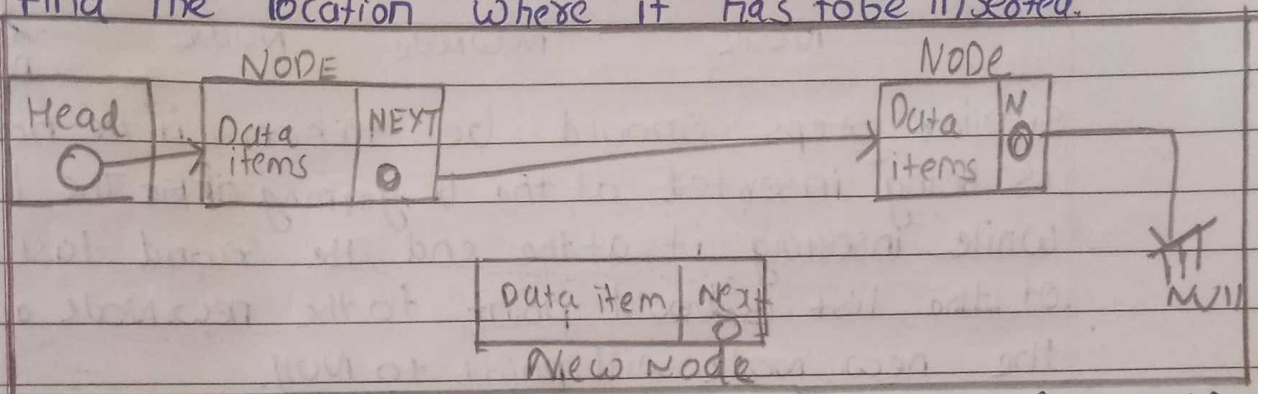
Following are the basic operations supported by a list

- Insertion :→ Adds an element at the beginning of the lists

- Deletion :→ Delets an element at the beginning of the list.
- Disply → Display the complete List.
- Search → Searches an element using the given key.
- Delete → Delets an element using the given key

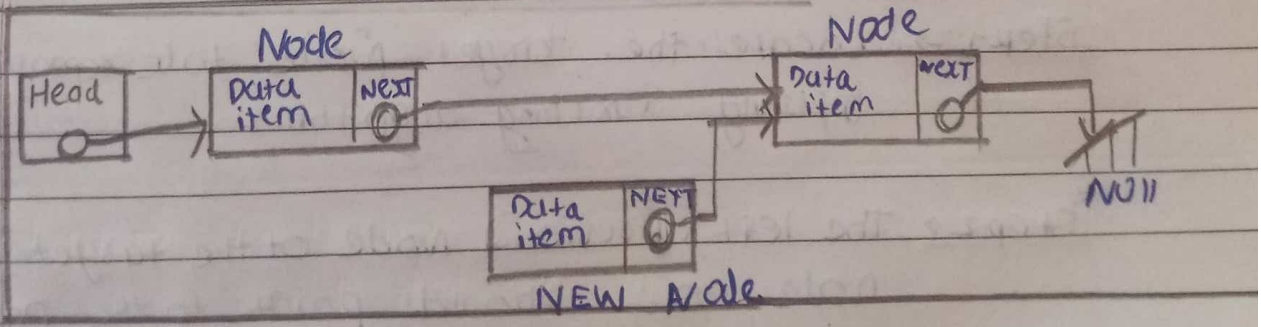
## Insertion operation

First, create a node using the same structure and Find the location where it has tobe inserted.



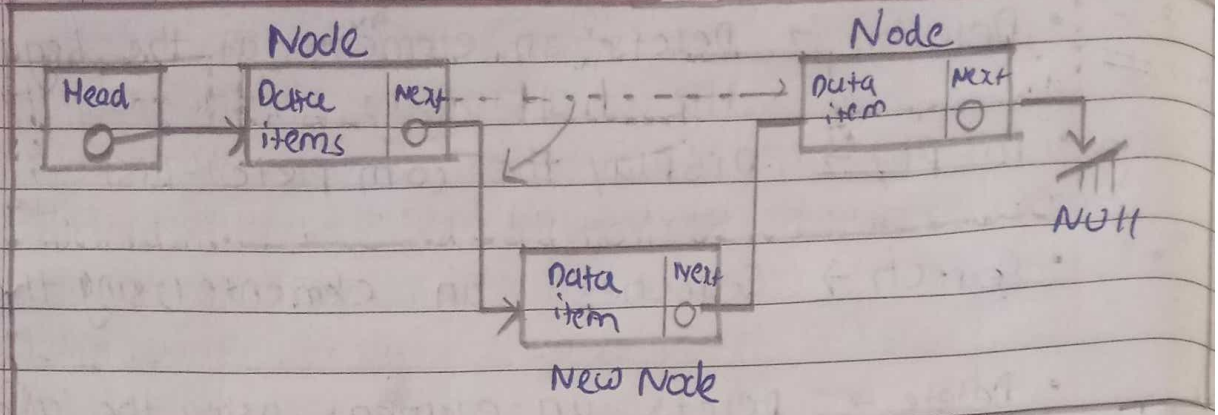

Imagin that we are inserting a node B (New Node), between A (Left Node) & C (Right Node)
Then point B.next to C-

```
NewNode.next → RightNode;
```

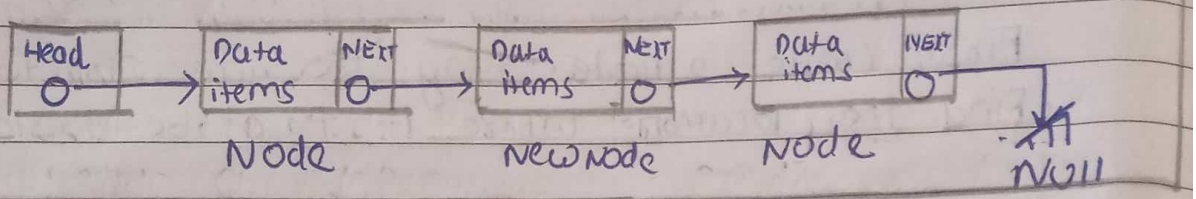



Now the next node at the Left should point to the New node

```
LeftNode.next → NewNode;
```




This will put New Node in middle of two Node




Similar steps should be taken if the node is being inserted at the beginning of the list. While inserting it at the end, the second last node of the list should point to the new Node and the new node will point to NULL.

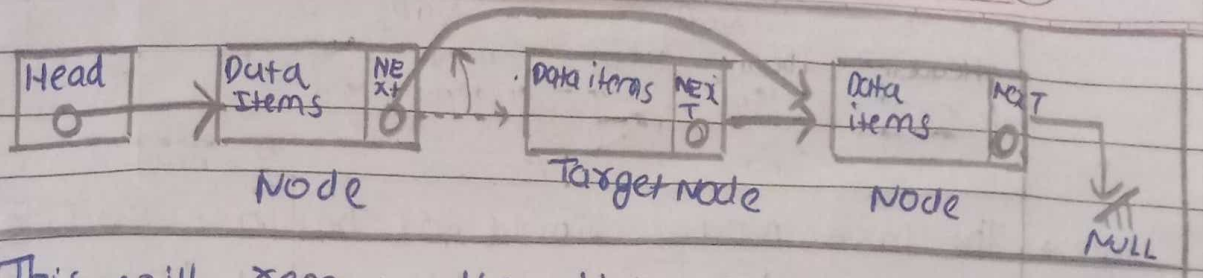
## ▶ Deletion Operation

Deletion is also a more than one step process. Pictorial representation

Step 1:→ Locate the target Node to be removed by using Searching algorithms.

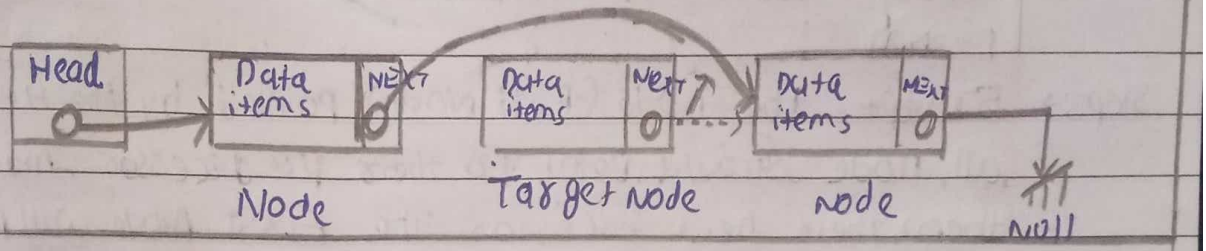
Step 2:→ The left (previous) Node of the target node now should point to the next node of the target Node-

```
LeftNode.next → TargetNode.next;
```

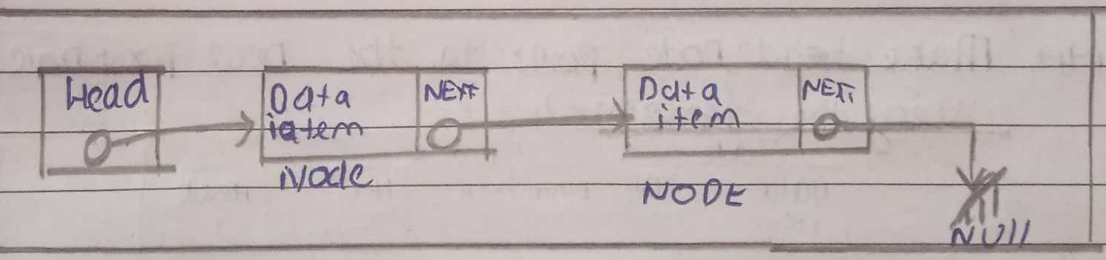



Step3:→ This will remove the link we pointed to the target Node. Now using the Following code, we will Remove what the target Node is pointing at.

```
TargetNode.next ->NULL;
```

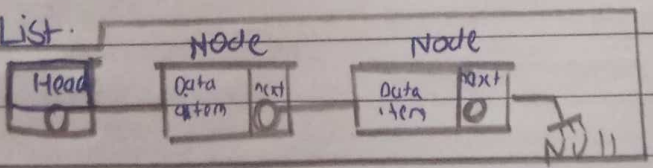



Step4:→ we need to use the deleted Node, we can keep that in memory otherwise we can simply de allocate memory and wipe off the target node completely

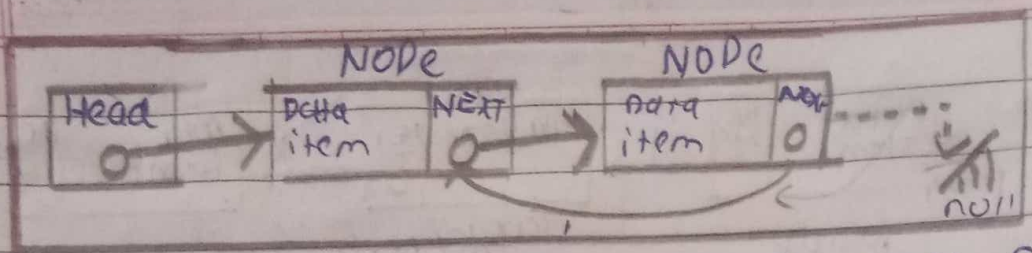



## Reverse Operation:→

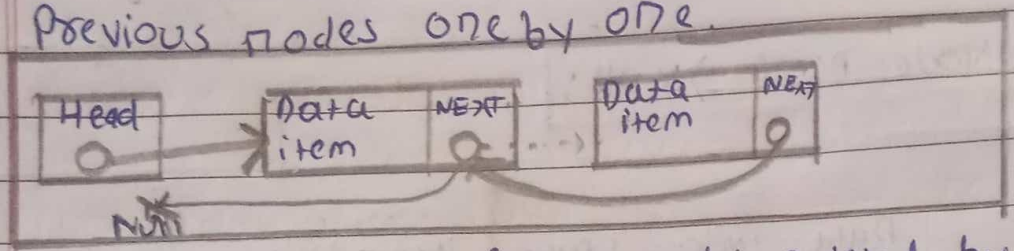
We need to point make the last node to be pointed by the head Node and reverse the whole linked List.



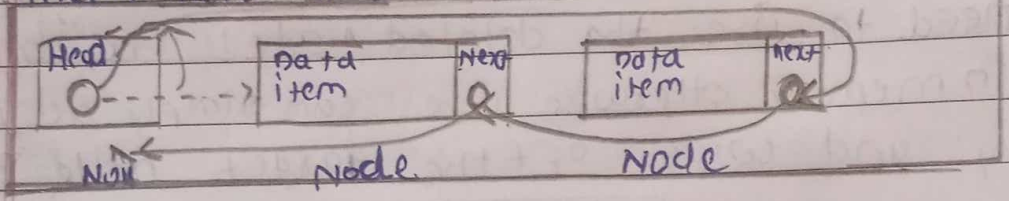

Step1:→ First we traverse to the end of the list. It should be pointing to NULL. Now, we shall make it point to its previous node-



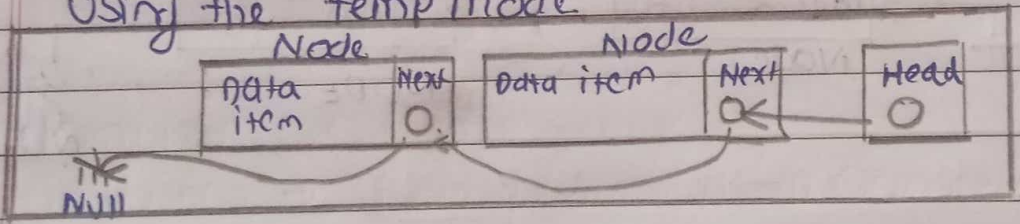

Step2:→ The last Node should not be lost. So we have add some temp Node ~~at the~~ which looks like the Head Node pointing to the last node.
Now, we will make all left side nodes point to their previous nodes one by one.




Step3:→ Except the node (First Node) pointed by the Head, node all node should point to their predecessor, making them their new successors. The First Node will point to NULL




Step4:→ Make Head node point to the new first node by using the temp node.




The linked list is reversed.

# Program code

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

struct node {
	string name;
	int MISno;
	node *next;
};
class member {
public:
	node *header1, *header2;
	member() {
		header1 = NULL;
		header2 = NULL;
	}

	node* create();
	int count(node *x);
	void add(node *head);
	void del(node *head);
	void display(node *head);
	void rdisplay(node *cn);
	void concatinate();
	// void option();
};
int member::count(node *x) {
```

```
	node *cn = x;
	int count = 0;
	while (cn != NULL) {
		count++;
		cn = cn->next;
	}
	return count;
}
node* member::create() {
	char ch;

	node *head;
	node *nn = new node;
	head = nn;
	cout << "Enter Name of president : ";
	cin >> nn->name;
	cout << "Enter MIS no. of president : ";
	cin >> nn->MISno;
	cout << endl;

	do {
		nn->next = new node;
		nn = nn->next;
		cout << "Enter Name of Member : ";
		cin >> nn->name;
		cout << "Enter MIS no. of Member : ";
		cin >> nn->MISno;
		cout << endl;
```

```
            cout << "Do you want to enter another member? (Y or y if yes) : ";
            cin >> ch;
            cout << endl;

        } while (ch == 'Y' || ch == 'y');

        nn->next = new node;

        nn = nn->next;

        cout << "Enter Name of secretary : ";
        cin >> nn->name;
        cout << "Enter MIS no. of secretary : ";
        cin >> nn->MISno;
        cout << endl;

        nn->next = NULL;

        cout << "List is created! It has " << count(head) << " members!" << endl<< endl;

        return head;
}
void member::display(node *head) {

        node *nn;
        nn = head;

        cout << "President : " << nn->name << endl;
        cout << "MIS no. : " << nn->MISno << endl << endl;
```

```
        nn = nn->next;



        for (int i = 0; nn -> next != NULL; i++) {
                cout << "Member : " << nn->name << endl;
                cout << "MIS no. : " << nn->MISno << endl << endl;
                nn = nn->next;
        }

        cout << "Secretary : " << nn->name << endl;
        cout << "MIS no. : " << nn->MISno << endl << endl;

        cout << "List has " << count(head) << " members!" << endl << endl;
}
void member::rdisplay(node *cn) {
        if (cn == NULL)
                return;

        rdisplay(cn->next);
        cout << cn->name << endl;
        cout << cn->MISno << endl;
        cout << endl;
}
void member::add(node *head) {
        int p;
        cout << "Enter the position where you want to add : ";
        cin >> p;

        node *nn, *temp;
```

```
nn = head;


node *an = new node;
if (p == 1) {
        cout << "Enter Name of Member : ";
        cin >> an->name;
        cout << "Enter MIS no. of Member : ";
        cin >> an->MISno;
        cout << endl;
        an->next = head;
        head = an;
        cout << "Member is added! List has " << count(head)
                        << " members now!\n\n";

} else if (p == count(head) + 1) {
        for (int i = 0; nn->next != NULL; i++) {
                nn = nn->next;
        }

        cout << "Enter Name of Member : ";
        cin >> an->name;
        cout << "Enter MIS no. of Member : ";
        cin >> an->MISno;
        cout << endl;
        nn->next = an;
        an->next = NULL;

        cout << "Member is added! List has " << count(head)
                        << " members now!\n\n";
```

```
                ;
        } else if (1 < p && p <= count(head)) {
                for (int i = 1; i < p; i++) {
                        temp = nn;
                        nn = nn->next;
                }

                cout << "Enter Name of Member : ";
                cin >> an->name;
                cout << "Enter MIS no. of Member : ";
                cin >> an->MISno;
                cout << endl;
                an->next = nn;
                temp->next = an;

                cout << "Member is added! List has " << count(head)
                                << " members now!\n\n";
        } else
                cout << "Invalid Position!\n\n";
}
void member::del(node *head) {
        int key;
        cout << "Enter the MIS no. of the student which is to be deleted : ";
        cin >> key;
        cout << endl;

        node *nn, *temp;
        nn = head;
```

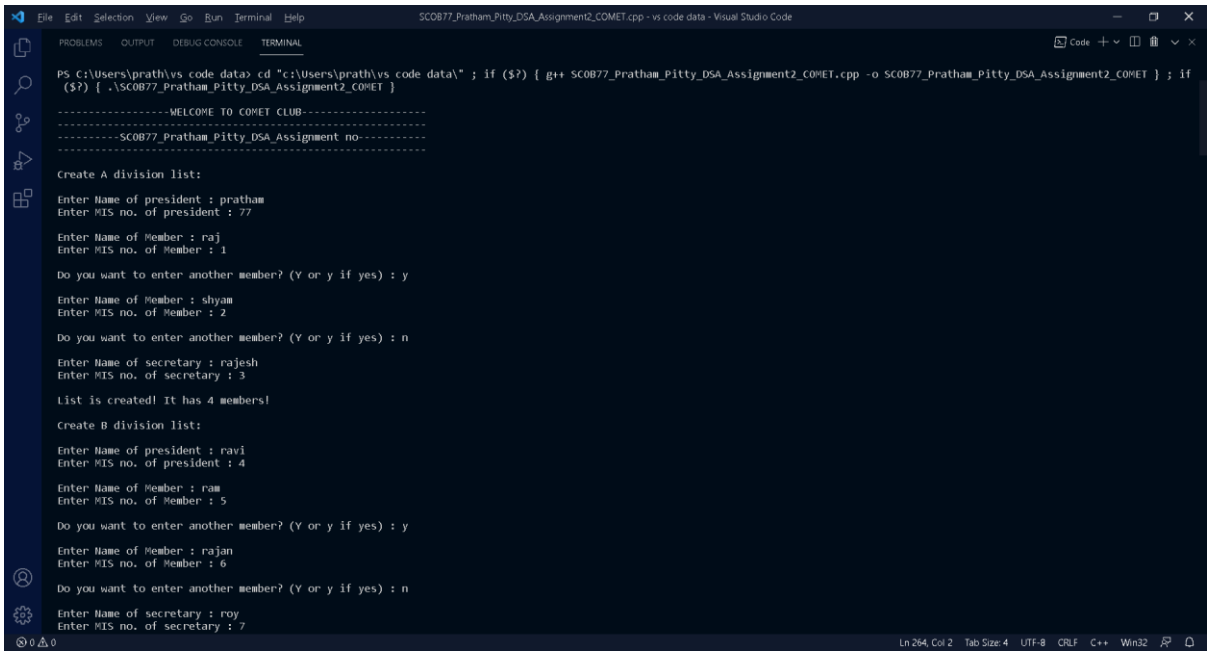
```
        if (nn->MISno == key) {
                head = nn->next;
                delete (nn);
                cout << "Member deleted! List has " << count(head)
                                << " members now!\n\n";
        } else {
                while (nn->MISno != key) {
                        temp = nn;
                        nn = nn->next;
                }

                if (nn->MISno == key && nn->next == NULL) {
                        temp->next = NULL;
                        delete (nn);
                        cout << "Member deleted! List has " << count(head)
                                        << " members now!\n\n";
                } else if (nn->MISno == key) {
                        temp->next = nn->next;
                        delete (nn);
                        cout << "Member deleted! List has " << count(head)
                                        << " members now!\n\n";
                } else
                        cout << "Member not found!\n\n";
        }
}
void member::concatinate()
{
        node *cn = header1;
        while(cn -> next != NULL)
```

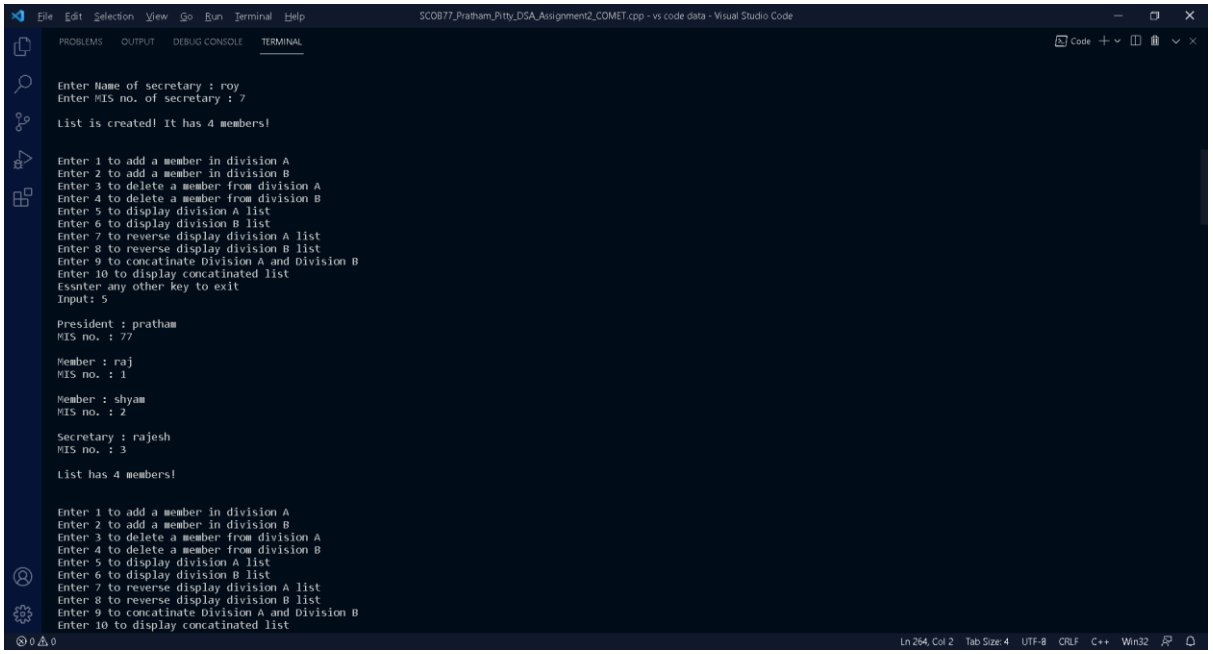
```
            cn = cn -> next;


        cn -> next = header2;


        cout << "Lists of Division A and Division B are concatinated! Club has total " <<
count(header1) << " members now!\n\n";
}
int main() {
        int choice,g,a;
        member m;
            cout << "\n------------------WELCOME TO COMET CLUB--------------------\n";
            cout <<"------------------------------------------------------------\n";
            cout <<"----------SCOB77_Pratham_Pitty_DSA_Assignment no-----------\n";
            cout <<"------------------------------------------------------------\n\n";
        cout << "Create A division list: " << endl << endl;
        m.header1 = m.create();
        cout << "Create B division list: " << endl << endl;
        m.header2 = m.create();



        while (true) {
            cout << endl
                                << "Enter 1 to add a member in division A\nEnter 2 to add a
member in division B\nEnter 3 to delete a member from division A\nEnter 4 to delete a
member from division B\nEnter 5 to display division A list\nEnter 6 to display division B
list\nEnter 7 to reverse display division A list\nEnter 8 to reverse display division B
list\nEnter 9 to concatinate Division A and Division B\nEnter 10 to display concatinated
list\nEssnter any other key to exit\nInput: ";
            cin >> choice;
            cout << endl;
```

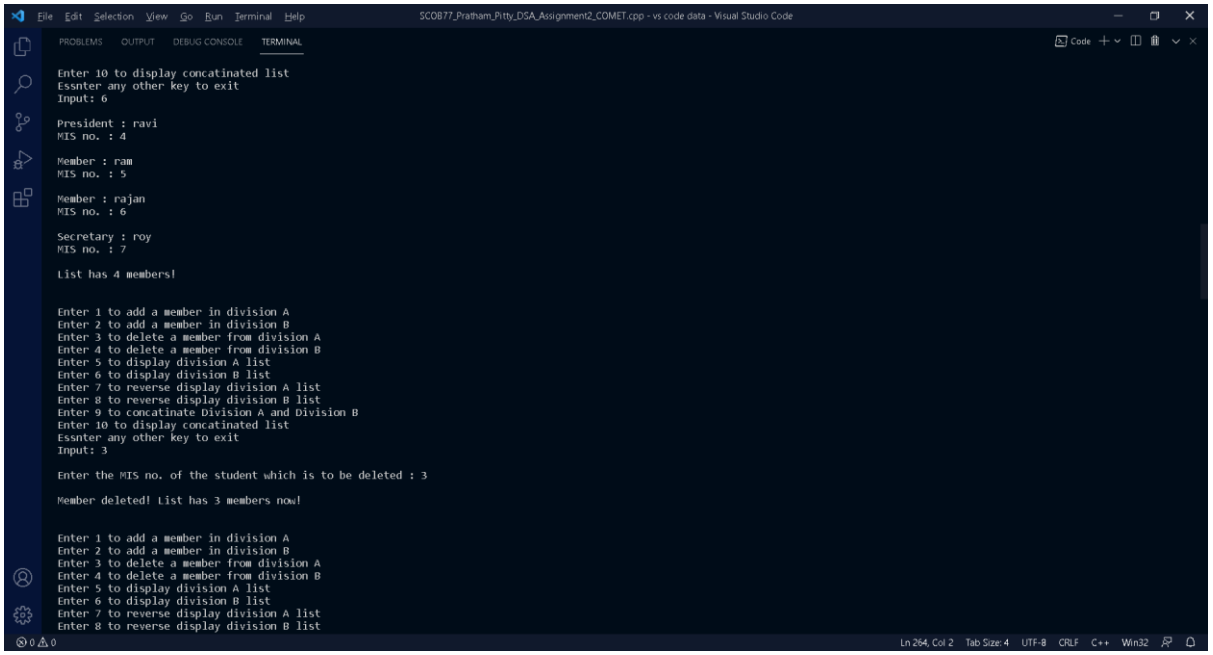
```
switch (choice) {


case 1:
        m.add(m.header1);
        break;
case 2:
        m.add(m.header2);
        break;
case 3:
        m.del(m.header1);
        break;
case 4:
        m.del(m.header2);
        break;
case 5:
        m.display(m.header1);
        break;
case 6:
        m.display(m.header2);
        break;
case 7:
        m.rdisplay(m.header1);
        break;
case 8:
        m.rdisplay(m.header2);
        break;
case 9:
        m.concatinate();
        break;
```

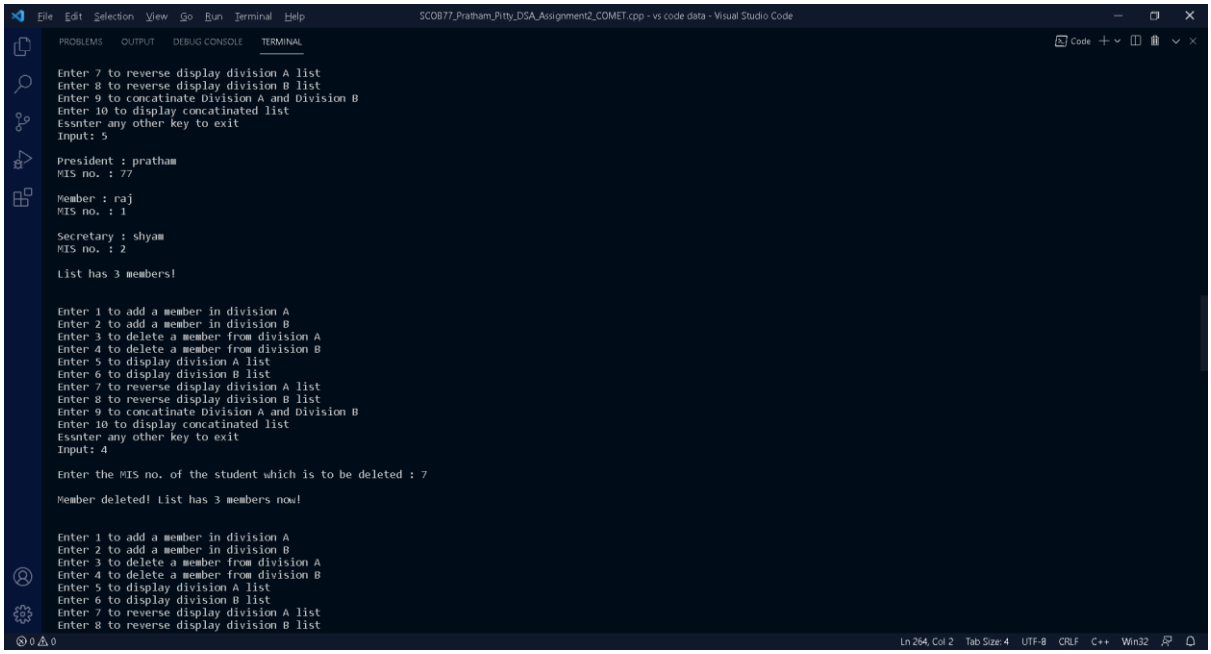
```
            case 10:
                m.display(m.header1);
                break;
            default:
                return 0;
            }
        }

}
```

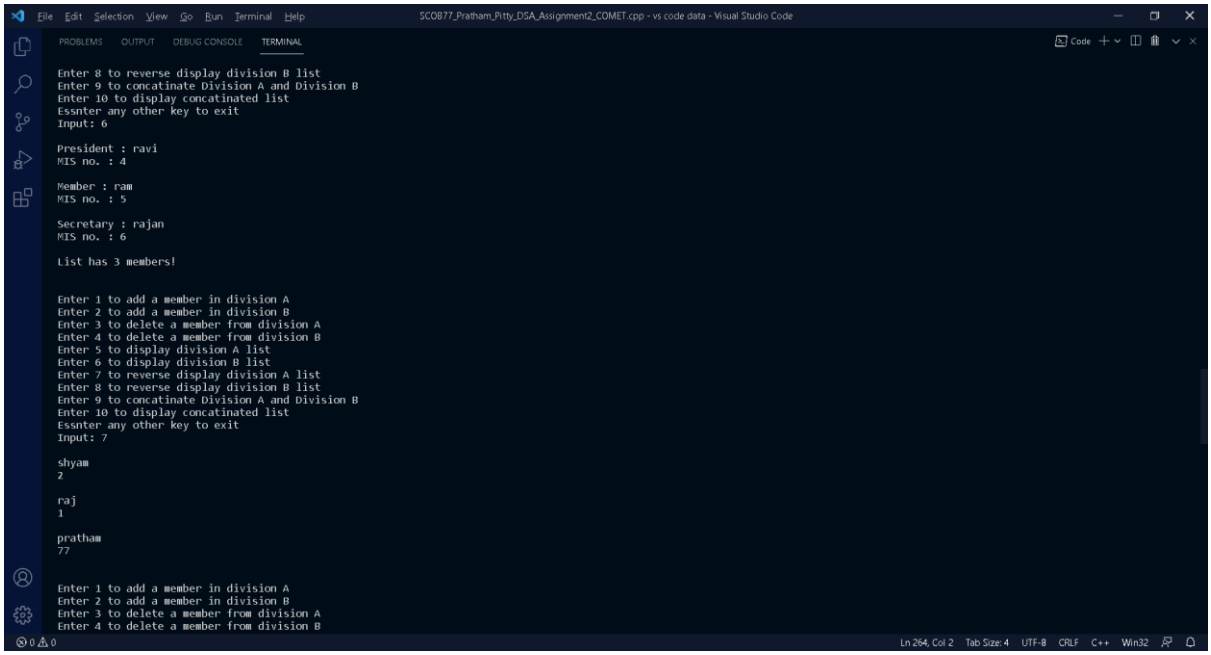
# Output:-

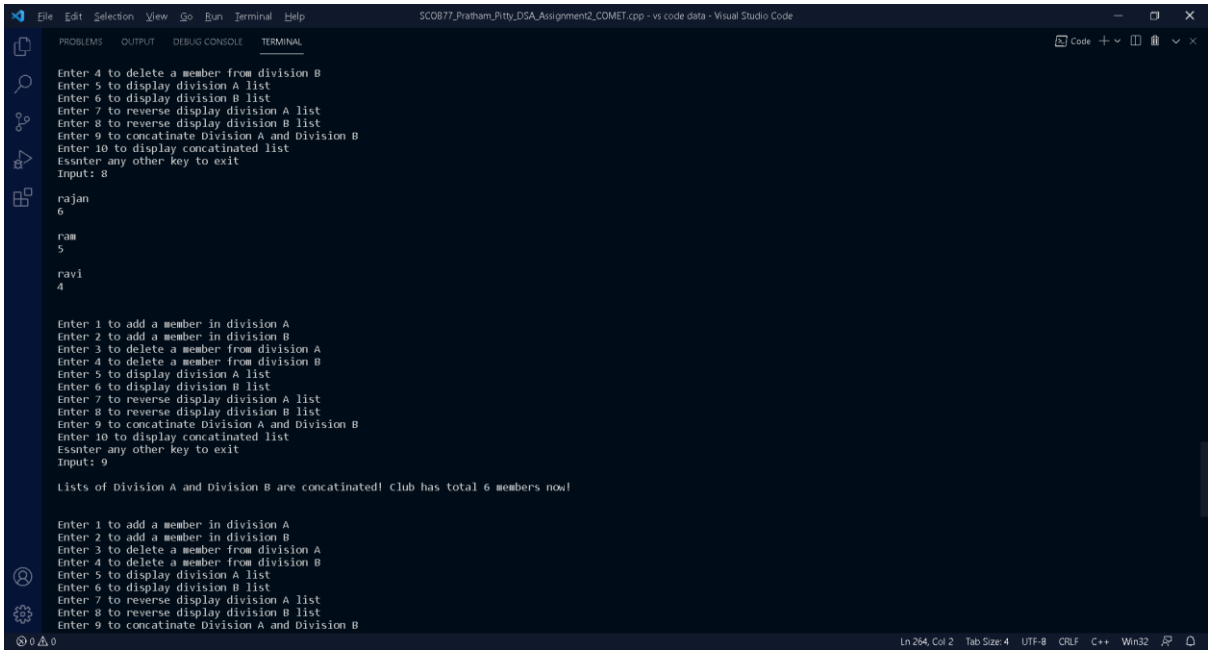


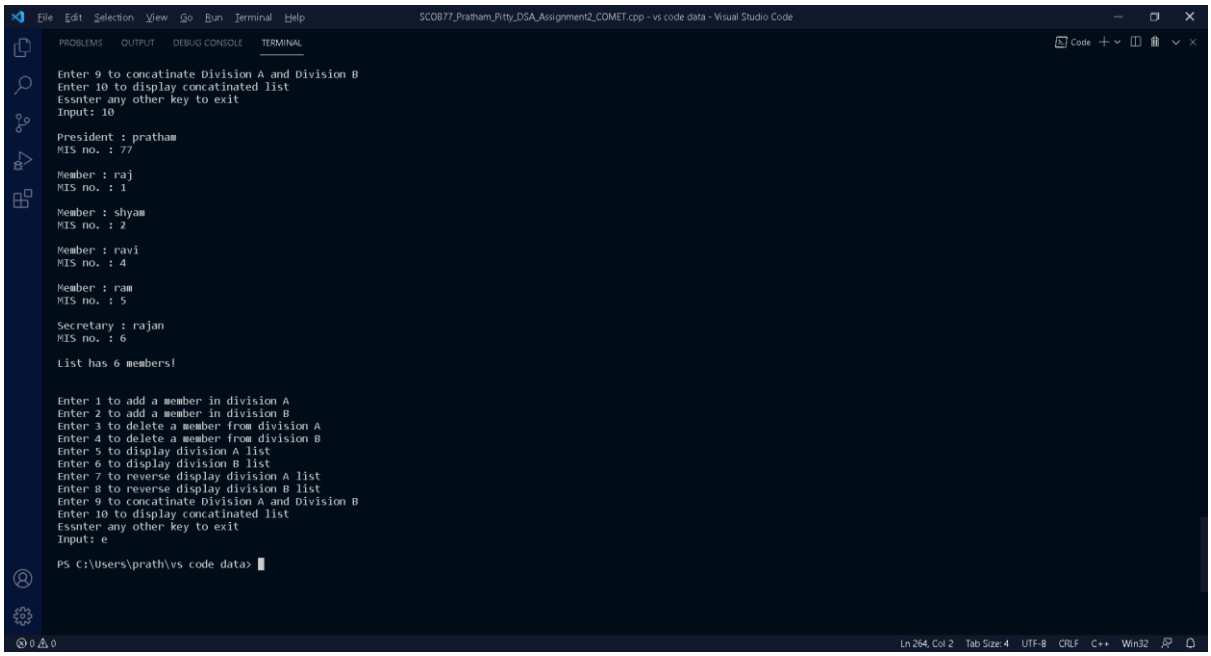

File Edit Selection View Go Run Terminal Help
SC0877_Pratham_Pitty_DSA_Assignment2_COMET.cpp - vs code data - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
Enter Name of secretary : roy
Enter MIS no. of secretary : 7
List is created! It has 4 members!
Enter 1 to add a member in division A
Enter 2 to add a member in division B
Enter 3 to delete a member from division A
Enter 4 to delete a member from division B
Enter 5 to display division A list
Enter 6 to display division B list
Enter 7 to reverse display division A list
Enter 8 to reverse display division B list
Enter 9 to concatinate Division A and Division B
Enter 10 to display concatinated list
Essnter any other key to exit
Input: 5
President : pratham
MIS no. : 77
Member : raj
MIS no. : 1
Member : shyam
MIS no. : 2
Secretary : rajesh
MIS no. : 3
List has 4 members!
Enter 1 to add a member in division A
Enter 2 to add a member in division B
Enter 3 to delete a member from division A
Enter 4 to delete a member from division B
Enter 5 to display division A list
Enter 6 to display division B list
Enter 7 to reverse display division A list
Enter 8 to reverse display division B list
Enter 9 to concatinate Division A and Division B
Enter 10 to display concatinated list
Ln 264, Col 2 Tab Size: 4 UTF-8 CRLF C++ Win32


File Edit Selection View Go Run Terminal Help
SC0877_Pratham_Pitty_DSA_Assignment2_COMET.cpp - vs code data - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
Enter 10 to display concatinated list
Essnter any other key to exit
Input: 6
President : ravi
MIS no. : 4
Member : ram
MIS no. : 5
Member : rajan
MIS no. : 6
Secretary : roy
MIS no. : 7
List has 4 members!
Enter 1 to add a member in division A
Enter 2 to add a member in division B
Enter 3 to delete a member from division A
Enter 4 to delete a member from division B
Enter 5 to display division A list
Enter 6 to display division B list
Enter 7 to reverse display division A list
Enter 8 to reverse display division B list
Enter 9 to concatinate Division A and Division B
Enter 10 to display concatinated list
Essnter any other key to exit
Input: 3
Enter the MIS no. of the student which is to be deleted : 3
Member deleted! List has 3 members now!
Enter 1 to add a member in division A
Enter 2 to add a member in division B
Enter 3 to delete a member from division A
Enter 4 to delete a member from division B
Enter 5 to display division A list
Enter 6 to display division B list
Enter 7 to reverse display division A list
Enter 8 to reverse display division B list
Ln 264, Col 2 Tab Size: 4 UTF-8 CRLF C++ Win32


File Edit Selection View Go Run Terminal Help
SC0877_Pratham_Pitty_DSA_Assignment2_COMET.cpp - vs code data - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
Enter 7 to reverse display division A list
Enter 8 to reverse display division B list
Enter 9 to concatinate Division A and Division B
Enter 10 to display concatinated list
Essnter any other key to exit
Input: 5
President : pratham
MIS no. : 77
Member : raj
MIS no. : 1
Secretary : shyam
MIS no. : 2
List has 3 members!
Enter 1 to add a member in division A
Enter 2 to add a member in division B
Enter 3 to delete a member from division A
Enter 4 to delete a member from division B
Enter 5 to display division A list
Enter 6 to display division B list
Enter 7 to reverse display division A list
Enter 8 to reverse display division B list
Enter 9 to concatinate Division A and Division B
Enter 10 to display concatinated list
Essnter any other key to exit
Input: 4
Enter the MIS no. of the student which is to be deleted : 7
Member deleted! List has 3 members now!
Enter 1 to add a member in division A
Enter 2 to add a member in division B
Enter 3 to delete a member from division A
Enter 4 to delete a member from division B
Enter 5 to display division A list
Enter 6 to display division B list
Enter 7 to reverse display division A list
Enter 8 to reverse display division B list
Ln 264, Col 2 Tab Size: 4 UTF-8 CRLF C++ Win32


File Edit Selection View Go Run Terminal Help
SC0877_Pratham_Pitty_DSA_Assignment2_COMET.cpp - vs code data - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
Enter 8 to reverse display division B list
Enter 9 to concatinate Division A and Division B
Enter 10 to display concatinated list
Essnter any other key to exit
Input: 6
President : ravi
MIS no. : 4
Member : ram
MIS no. : 5
Secretary : rajan
MIS no. : 6
List has 3 members!
Enter 1 to add a member in division A
Enter 2 to add a member in division B
Enter 3 to delete a member from division A
Enter 4 to delete a member from division B
Enter 5 to display division A list
Enter 6 to display division B list
Enter 7 to reverse display division A list
Enter 8 to reverse display division B list
Enter 9 to concatinate Division A and Division B
Enter 10 to display concatinated list
Essnter any other key to exit
Input: 7
shyam
2
raj
1
pratham
77
Enter 1 to add a member in division A
Enter 2 to add a member in division B
Enter 3 to delete a member from division A
Enter 4 to delete a member from division B
Ln 264, Col 2 Tab Size: 4 UTF-8 CRLF C++ Win32


File Edit Selection View Go Run Terminal Help
SC0877_Pratham_Pitty_DSA_Assignment2_COMET.cpp - vs code data - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
Enter 4 to delete a member from division B
Enter 5 to display division A list
Enter 6 to display division B list
Enter 7 to reverse display division A list
Enter 8 to reverse display division B list
Enter 9 to concatinate Division A and Division B
Enter 10 to display concatinated list
Essnter any other key to exit
Input: 8
rajan
6
ram
5
ravi
4
Enter 1 to add a member in division A
Enter 2 to add a member in division B
Enter 3 to delete a member from division A
Enter 4 to delete a member from division B
Enter 5 to display division A list
Enter 6 to display division B list
Enter 7 to reverse display division A list
Enter 8 to reverse display division B list
Enter 9 to concatinate Division A and Division B
Enter 10 to display concatinated list
Essnter any other key to exit
Input: 9
Lists of Division A and Division B are concatinated! Club has total 6 members now!
Enter 1 to add a member in division A
Enter 2 to add a member in division B
Enter 3 to delete a member from division A
Enter 4 to delete a member from division B
Enter 5 to display division A list
Enter 6 to display division B list
Enter 7 to reverse display division A list
Enter 8 to reverse display division B list
Enter 9 to concatinate Division A and Division B
Ln 264, Col 2 Tab Size: 4 UTF-8 CRLF C++ Win32


File Edit Selection View Go Run Terminal Help
SC0877_Pratham_Pitty_DSA_Assignment2_COMET.cpp - vs code data - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
Enter 9 to concatinate Division A and Division B
Enter 10 to display concatinated list
Essnter any other key to exit
Input: 10
President : pratham
MIS no. : 77
Member : raj
MIS no. : 1
Member : shyam
MIS no. : 2
Member : ravi
MIS no. : 4
Member : ram
MIS no. : 5
Secretary : rajan
MIS no. : 6
List has 6 members!
Enter 1 to add a member in division A
Enter 2 to add a member in division B
Enter 3 to delete a member from division A
Enter 4 to delete a member from division B
Enter 5 to display division A list
Enter 6 to display division B list
Enter 7 to reverse display division A list
Enter 8 to reverse display division B list
Enter 9 to concatinate Division A and Division B
Enter 10 to display concatinated list
Essnter any other key to exit
Input: e
PS C:\Users\prath\vs code data>
Ln 264, Col 2 Tab Size: 4 UTF-8 CRLF C++ Win32