```cpp
#include<iostream>
using namespace std;
class sll
{
    struct node
    {
        int prn;
        char name[10];
        node *next;
    }*start;
    public:
        sll()
        {
            start=NULL;
        }
        void create();
        void display();
        void insert_beginning();  // president
        void insert_end();        // secretary
        void insert_mid();       // members
        void del_beginning();
        void del_end();
        void del_mid();
        void compute();
        void concatenate(sll obj2);
};
void sll::create()
{   node *temp;
    node *curr;
    int prn;
    char name;
    int ans;
    do
    {
        temp = new node;
        temp->next=NULL;
        cout<<"Enter PRN number :"<<endl;
        cin>>temp->prn;
        cout<<"Enter name: "<<endl;
        cin>>temp->name;

        if (start==NULL)
        {
            start=temp;
            curr = temp;
        }
        else
        {
            curr->next = temp;
            curr = temp;
        }
        cout<<"Do you want to add a new node ? 1 for yes"<<endl;
        cin>>ans;

    }while(ans==1);
}
void sll::display()
{
    node *t;
    if(start==NULL)
    {
        cout<<"Club is empty"<<endl;
        return;
    }
    else
    {
        t = start;
        while(t!=NULL)
        {
            cout<<t->prn<<" "<<t->name<<"->";
            t = t->next;
        }
        cout<<"NULL";
    }
}
void sll::insert_beginning()
{
    node *temp;
    temp = new node;
    temp->next = NULL;
    cout<<"Enter PRN number: "<<endl;
    cin>>temp->prn;
    cout<<"Enter name: "<<endl;
    cin>>temp->name;
    if (start == NULL)
    {
        start = temp;
    }
    else
    {
        temp->next = start;
        start = temp;
    }
}
void sll::insert_end()
{
    node *temp, *last;
    temp = new node;
    temp->next = NULL;
    cout<<"Enter PRN number: "<<endl;
    cin>>temp->prn;
    cout<<"Enter name: "<<endl;
    cin>>temp->name;
    if(start == NULL)
    {
        start = temp;
    }
    else
    {
        last = start;
        while (last->next != NULL)
        {
            last = last->next;
        }
        last->next=temp;
    }
}
void sll::insert_mid()
{
    node *temp;
    node *curr;
    int loc;
    cout<<"\nEnter location after which you want to insert: "<<endl;
    cin>>loc;
    temp=new node;
    temp->next=NULL;
    cout<<"Enter PRN number: "<<endl;
    cin>>temp->prn;
    cout<<"Enter name: "<<endl;
    cin>>temp->name;
    curr=start;
    for(int i=1;i<loc;i++)
    {
        curr=curr->next;
    }
    temp->next=curr->next;  //right side link of new node temp
    curr->next=temp;        //left side link of new node temp
}
void sll::del_beginning()
{
    node *temp;
    if(start==NULL)
    {
        cout<<"\nClub is empty"<<endl;
    }
    else
    {
    temp=start;
    start=start->next;
    cout<<temp->prn<<"\t first element deleted"<<endl;
    delete temp;

    }
}
void sll::del_end()
{
    node *temp,*prev;
    if(start==NULL)
    {
        cout<<"\nClub is empty"<<endl;
    }
    else
    {
        temp=start;
    while(temp->next!=NULL)
    {
        prev=temp;
        temp=temp->next;
    }

    cout<<temp->prn<<"\t last element deleted."<<endl;;
    delete temp;
    prev->next=NULL;

    }
}
void sll::del_mid()
{
    node *temp;
    node *curr;
    int loc;
    cout<<"\nEnter location of the node which you want to delete: "<<endl;
    cin>>loc;
    curr=start;
    for(int i=1;i<loc;i++)
    {
        temp=curr;
        curr=curr->next;
    }
    temp->next=curr->next;
    cout<<curr->prn<<"\t has been deleted"<<endl;
    delete curr;
}
void sll::compute()
{
    node *temp;
    int count=0;
    if(start==NULL)
    {
        cout<<"\nClub is empty"<<endl;
    }
    temp=start;
    while(temp!=NULL)
    {
        count++;
        temp=temp->next;
    }
    cout<<"Total no of members are\t"<<count<<endl;
}
void sll::concatenate(sll obj2)
{
    node *temp,*last;
    last=obj2.start;
    if(last==NULL)
    {
        cout<<"\nList 2 is empty"<<endl;
        return;
    }
    temp=start;
    while(temp->next!=NULL)
    {
        temp=temp->next;
    }
    temp->next=last;
    cout<<"\nAfter concatenation: ";
}

int main()
{
    sll obj;
    int ch;
    do
    {
        cout<<"\n1. create\n2.Insert at beginning\n3.Insert at end\n4.insert after
position\n5.Display list\n6.Delete first element\n7.Delete last element\n8.Delete
Member\n9.Find total No. of members\n10.Concatenate lists\n0. Exit\nEnter your choice: ";
        cin>>ch;
        switch(ch)
        {
            case 1:obj.create();
                   obj.display();
                   break;
            case 2:obj.insert_beginning();
                   obj.display();
                   break;
            case 3:obj.insert_end();
                   obj.display();
                   break;
            case 4:obj.insert_mid();
                   obj.display();
                   break;
            case 5:obj.display();
                   break;
            case 6:obj.del_beginning();
                   obj.display();
                   break;
            case 7:obj.del_end();
                   obj.display();
                   break;
            case 8:obj.del_mid();
                   obj.display();
                   break;
            case 9:obj.compute();
                   obj.display();
                   break;
            case 10:
            {
                sll obj2, obj3;
                cout<<"\nList 1: "<<endl;
                obj2.create();
                cout<<"\nList 2: "<<endl;
                obj3.create();
                obj2.concatenate(obj3);
                obj2.display();
                break;
            }
        }
    }while(ch!=0);
cout<<"\nEnd of program"<<endl;
    return 0;
}

/
******************************************************************************************/

Output : -

ubuntu@ubuntu-Vostro-460:~$ g++ practical7dsl.cpp
ubuntu@ubuntu-Vostro-460:~$ ./a.out

1. create
2.Insert at beginning
3.Insert at end
4.insert after position
5.Display list
6.Delete first element
7.Delete last element
8.Delete Member
9.Find total No. of members
10.Concatenate lists
0. Exit
Enter your choice: 1
Enter PRN number :
12345678
Enter name:
Brandon
Do you want to add a new node ? 1 for yes
1
Enter PRN number :
123456789
Enter name:
Shannon
Do you want to add a new node ? 1 for yes
0
12345678 Brandon->123456789 Shannon->NULL
1. create
2.Insert at beginning
3.Insert at end
4.insert after position
5.Display list
6.Delete first element
7.Delete last element
8.Delete Member
9.Find total No. of members
10.Concatenate lists
0. Exit
Enter your choice: 5
12345678 Brandon->123456789 Shannon->NULL
1. create
2.Insert at beginning
```

```
3.Insert at end
4.insert after position
5.Display list
6.Delete first element
7.Delete last element
8.Delete Member
9.Find total No. of members
10.Concatenate lists
0. Exit
Enter your choice: 6
12345678     first element deleted
123456789 Shannon->NULL
1. create
2.Insert at beginning
3.Insert at end
4.insert after position
5.Display list
6.Delete first element
7.Delete last element
8.Delete Member
9.Find total No. of members
10.Concatenate lists
0. Exit
Enter your choice: 2
Enter PRN number:
12345678
Enter name:
Karen
12345678 Karen->123456789 Shannon->NULL
1. create
2.Insert at beginning
3.Insert at end
4.insert after position
5.Display list
6.Delete first element
7.Delete last element
8.Delete Member
9.Find total No. of members
10.Concatenate lists
0. Exit
Enter your choice: 7
123456789    last element deleted.
12345678 Karen->NULL
1. create
2.Insert at beginning
3.Insert at end
4.insert after position
5.Display list
6.Delete first element
7.Delete last element
8.Delete Member
9.Find total No. of members
10.Concatenate lists
0. Exit
Enter your choice: 9
Total no of members are 1
12345678 Karen->NULL
1. create
2.Insert at beginning
3.Insert at end
4.insert after position
5.Display list
6.Delete first element
7.Delete last element
8.Delete Member
9.Find total No. of members
```

```
10.Concatenate lists
0. Exit
Enter your choice: 0

End of program
```