```
#include<stdio.h>
#include<stdlib.h>

struct node
{
    char usn[25],name[25],branch[25];
    int sem;
    long int phone;
    struct node *link;
};
typedef struct node * NODE;

NODE start = NULL;
int count=0;


NODE create()
{
    NODE snode;
    snode = (NODE)malloc(sizeof(struct node));

    if(snode == NULL)
    {
      printf("\nMemory is not available");
      exit(1);
    }
    printf("\nEnter the usn,Name,Branch, sem,PhoneNo of the student:");
    scanf("%s %s %s %d %ld",snode->usn, snode->name, snode->branch,
&snode->sem, &snode->phone);
    snode->link=NULL;
    count++;
    return snode;
}

NODE insertfront()
{
    NODE temp;
    temp = create();
    if(start == NULL)
    {
         return temp;
    }

    temp->link = start;
    return temp;
}


NODE deletefront()
{
    NODE temp;
    if(start == NULL)
    {
      printf("\nLinked list is empty");
```

```
      return NULL;
    }

    if(start->link == NULL)
    {
          printf("\nThe Student node with usn:%s is deleted ",start-
>usn);
          count--;
          free(start);
          return NULL;
    }
    temp = start;
    start = start->link;
    printf("\nThe Student node with usn:%s is deleted",temp->usn);
    count--;
    free(temp);
    return start;
}

NODE insertend()
{
    NODE cur,temp;
    temp = create();

    if(start == NULL)
    {
      return temp;
    }
    cur = start;
    while(cur->link !=NULL)
    {
       cur = cur->link;
    }
    cur->link = temp;
    return start;
}

NODE deleteend()
{
     NODE cur,prev;
     if(start == NULL)
     {
      printf("\nLinked List is empty");
      return NULL;
     }

     if(start->link == NULL)
     {
      printf("\nThe student node with the usn:%s is deleted",start->usn);
      free(start);
      count--;
      return NULL;
     }
```

```
     prev = NULL;
     cur = start;
     while(cur->link!=NULL)
     {
       prev = cur;
       cur = cur->link;
     }

      printf("\nThe student node with the usn:%s is deleted",cur->usn);
      free(cur);
      prev->link = NULL;
      count--;
      return start;
}

void display()
{
    NODE cur;
    int num=1;


    if(start == NULL)
    {

      printf("\nNo Contents to display in SLL \n");
      return;
    }
    printf("\nThe contents of SLL: \n");
    cur = start;
    printf("\nSNo.\t USN\t Name\t Branch\t Sem\t PhNo\n");
    while(cur!=NULL)
    {
       printf("\n %d\t %s\t %s\t %s\t %d\t %ld\n",num,cur->usn, cur-
>name,cur->branch, cur->sem,cur->phone);
       cur = cur->link;
       num++;
    }
    printf("\n No of student nodes is %d \n",count);
}

void stackdemo()
{
   int ch;
   while(1)
   {
     printf("\n----------Stack Demo using SLL------------\n");
     printf("\n1:Push operation \n2: Pop operation \n3: Display \n4:Exit
\n");
     printf("\nEnter your choice for stack demo");
     scanf("%d",&ch);

     switch(ch)
     {
      case 1: start = insertfront();
```

```
            break;
      case 2: start = deletefront();
            break;
      case 3: display();
             break;
       default : return;
      }
   }

}

void main()
{
    int ch,i,n;
    while(1)
    {
      printf("\n~~~Menu~~~");
      printf("\nEnter your choice for SLL operation \n");
      printf("\n1:Create SLL of Student Nodes");
      printf("\n2:DisplayStatus");
      printf("\n3:InsertAtEnd");
      printf("\n4:DeleteAtEnd");
      printf("\n5:Stack Demo using SLL(Insertion and Deletion at
Front)");
      printf("\n6:Exit \n");
      printf("\nEnter your choice:");
      scanf("%d",&ch);

      switch(ch)
      {
      case 1 : printf("\nEnter the no of students:    ");
             scanf("%d",&n);
             for(i=1;i<=n;i++)
                start = insertfront();
             break;

      case 2: display();
            break;

      case 3: start = insertend();
            break;

      case 4: start = deleteend();
            break;

      case 5: stackdemo();
            break;

      case 6: exit(0);

      default: printf("\nPlease enter the valid choice");

      }
    }
```

```
}
```