

## LAB PROGRAM 7

/\*\*\*\*\*\*

Design, Develop and Implement a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: *USN, Name, Branch, Sem, PhNo*

- Create a SLL of N Students Data by using *front insertion*.
- Display the status of SLL and count the number of nodes in it
- Perform Insertion and Deletion at End of SLL
- Perform Insertion and Deletion at Front of SLL
- Demonstrate how this SLL can be used as STACK and QUEUE
- Exit

\*\*\*\*\*/

```
#include <stdio.h>
#include<stdlib.h>
#include<string.h>
int count=0;
struct stud
{
    long int ph;
    int sem;
    char name[15],usn[15],brnch[8];
    struct stud *next;
} *head=NULL,*tail=NULL,*temp=NULL,*temp1;

void create(long int n,int s,char na[20],char u[15],char b[5])
{
    if(head==NULL)
    {
        head=(struct stud*)malloc(1*sizeof(struct stud));
        head->ph=n;
        head->sem=s;
        strcpy(head->name,na);
        strcpy(head->usn,u);
        strcpy(head->brnch,b);
        head->next=NULL;
        tail=head;
        count++;
    }
    else
    {
        temp=(struct stud*)malloc(1*sizeof(struct stud));
        temp->ph=n;
        temp->sem=s;
        strcpy(temp->name,na);
        strcpy(temp->usn,u);
        strcpy(temp->brnch,b);
        temp->next=head;
        head=temp;
        count++;
    }
}
```

```

void display()
{
temp1=head;
if(temp1==NULL)
{
printf("\nlist is empty\n");
}
else
{
printf("student details are as follows:\n");
while(temp1!=NULL)
{
printf("-----\n");
printf("NAME:%s\nUSN:%s\nBRANCH:%s\nSEM:%d\nPHONE NO.:%ld\n",temp1-
>name,temp1->usn,temp1->brnch,temp1->sem,temp1->ph);
printf("-----\n");
temp1=temp1->next;
}
printf("no. of nodes=%d\n",count);
}
}

```

```

void insert_head(long int n,int s,char na[15],char u[15],char b[8])
{
temp=(struct stud*)malloc(1*sizeof(struct stud));
temp->ph=n;
temp->sem=s;
strcpy(temp->name,na);
strcpy(temp->usn,u);
strcpy(temp->brnch,b);
temp->next=head;
head=temp;
count++;
}

```

```

void insert_tail(long int n,int s,char na[15],char u[15],char b[8])
{
temp=(struct stud*)malloc(1*sizeof(struct stud));
temp->ph=n;
temp->sem=s;
strcpy(temp->name,na);
strcpy(temp->usn,u);
strcpy(temp->brnch,b);
tail->next=temp;
temp->next=NULL;
tail=temp;
count++;
}

```

```

void delete_head()
{
temp1=head;
if(temp1==NULL)
{
printf("list is empty\n");
}
else
{
head=head->next;
printf("deleted node is:\n");
printf("-----\n");
printf("NAME:%s\nUSN:%s\nBRANCH:%s\nSEM:%d\nPHONE NO.:%ld\n",temp1-
>name,temp1->usn,temp1->brnch,temp1->sem,temp1->ph);
printf("-----\n");
free(temp1);
count--;
}
}

void delete_tail()
{
    if(tail==NULL)
    {
printf("list is empty\n");
}
else if(head==tail)
{
printf("deleted node is:\n");
printf("-----\n");
printf("NAME:%s\nUSN:%s\nBRANCH:%s\nSEM:%d\nPHONE NO.:%ld\n",tail->name,tail-
>usn,tail->brnch,tail->sem,tail->ph);
printf("-----\n");
free(head);
tail=head=NULL;
count--;
}
else
{
temp1=head;
while(temp1->next!=tail)
{
temp1=temp1->next;
}
printf("deleted node is:\n");
printf("-----\n");
printf("NAME:%s\nUSN:%s\nBRANCH:%s\nSEM:%d\nPHONE NO.:%ld\n",tail->name,tail-
>usn,tail->brnch,tail->sem,tail->ph);
printf("-----\n");
free(tail);
}
}

```

```

tail=temp1;
tail->next=NULL;
count--;
}
}

void main()
{
int choice;
long int ph;
int sem;
char name[20],usn[15],brnch[5];
printf("-----MENU-----\n");
printf("1.create\n2.Insert from head\n3.Insert from tail\n4.Delete from head\n5.Delete from
tail\n6.display\n7.exit\n");
printf("-----\n");
while(1)
{
printf("enter your choice\n");
scanf("%d",&choice);
switch(choice)
{
case 1:printf("enter the name usn branch sem phno. of the student respectively\n");
scanf("%s%s%s%d%ld",name,usn,brnch,&sem,&ph);
create(ph,sem,name,usn,brnch);
break;
case 2: printf("enter the name usn branch sem phno. of the student respectively\n");
scanf("%s%s%s%d%ld",name,usn,brnch,&sem,&ph);
insert_head(ph,sem,name,usn,brnch);
break;
case 3: printf("enter the name usn branch sem phno. of the student respectively\n");
scanf("%s%s%s%d%ld",name,usn,brnch,&sem,&ph);
insert_tail(ph,sem,name,usn,brnch);
break;
case 4:delete_head();
break;
case 5:delete_tail();
break;
case 6:display();
break;
case 7: exit(0);
default:printf("invalid option\n");
}
}
}
}

```