```
/*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
      a. Create a SLL of N Students Data by using front insertion.
      b. Display the status of SLL and count the number of nodes in it
      c. Perform Insertion / Deletion at End of SLL
      d. Perform Insertion / Deletion at Front of SLL(Demonstration of
stack)
      e. Exit*/

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

struct NODE             /* NODE structure for Student Data*/
{
    int sem;
    long long phno;
    char name[20],branch[10],usn[20];
    struct NODE *next;  /* Node Link to next Student Data*/
};

typedef struct NODE *NODEPTR; /* Node Pointer for NODE structure*/
NODEPTR first = NULL;         /* first Pointer for LIST*/
int count=0;


NODEPTR create_node()
{
    NODEPTR node = (NODEPTR)malloc(sizeof(struct NODE)); /* Create NODE
dynamically*/
    printf("Enter the Student USN\n");
    scanf("%s",node->usn);
    printf("Enter the Student Name\n");
    scanf("%s",node->name);
    printf("Enter the Student BRANCH\n");
    scanf("%s",node->branch);
    printf("Enter the Student SEMESTER\n");
    scanf("%d",&node->sem);
    printf("Enter the Student PHONE NO.\n");
    scanf("%lld",&node->phno);
    node->next = NULL;  /* Initially NODE link is set to NULL*/
    count++;            /* Increment count when NODE i created*/
    return node;        /* return NODE Pointer */
}

void insert_front() /* Function to insert NODE to front */
{
    NODEPTR temp = create_node();
    temp->next = first;
    first=temp;
}
void delete_front()
{
    NODEPTR temp;
    temp = first;
    if(temp != NULL)
    {
        first = temp->next;
        temp->next = NULL;
        free(temp);
        count--;
        printf("Front node deleted successfully\n");
    }
    else
        printf("ALERT!!!:List is Empty\n");
}
void insert_end()
{
    NODEPTR last;
    NODEPTR temp = create_node();
    last = first;

        while(last->next != NULL)
        {
            last = last->next;
        }
        last->next = temp;

}

void delete_end()
{
    NODEPTR last, prev;
    last = first;
    if(first->next == NULL)
    {
        first = NULL;
        free(last);
        count--;
        printf("End node deleted successfully\n");
    }
    if(last != NULL)
    {
        while(last->next != NULL)
        {
            prev = last;
            last = last->next;
        }
        prev->next = NULL;
        free(last);
        count--;
        printf("End node deleted successfully\n");
    }
    else
        printf("ALERT!!!:List is Empty\n");
}

void display()
{
    NODEPTR temp;
    temp = first;
    if(first == NULL)
    {
        printf("List is Empty\n");
    }
    else
    {
        printf("The List values are ....\n");
        printf("[USN, Name, Branch, Sem, Phone]\n");
        while(temp!= NULL)
        {
            printf("[%s,%s,%s,%d,%lld]-->",temp->usn,temp->name,temp-
>branch,temp->sem,temp->phno);
            temp = temp->next;
        }
        printf("\nNODE COUNT = %d\n",count);
    }

}

int main()
{
      int  n, ch, i;
      while (1)
      {
            printf("\n**********Singly Linked List Operations
Menu**********\n");
            printf("1. Create a SLL of N Students Data by using front
insertion\n");
            printf("2. Display the status of SLL\n");
            printf("3. Insertion / Deletion at End of SLL\n");
            printf("4. Insertion / Deletion at Front of SLL\n");
            printf("5. Exit\n");
            printf("Enter your choice:\n");
            scanf("%d", &ch);
            switch (ch)
            {
            case 1: printf("Enter the value of N to create SLL\n");
                scanf("%d", &n);
                for(i=1;i<=n;i++)
                {
                    printf("Enter a %d node to insert towards front of
SLL\n",i);
                    insert_front();
                }
                break;
            case 2: display();
                       break;
            case 3: printf("Press 1 to Insert End or 2 to Delete End\n");
                scanf("%d", &ch);
                if(ch == 1)
                {
                    printf("Enter a node to insert towards end of
SLL\n");
                    insert_end();
                }
                else if(ch == 2)
                {
                    delete_end();
                }
                else
```

```
                    printf("Invalid Entry\n");
                break;
            case 4: printf("Press 1 to Insert Front or 2 to Delete 
Front\n");
                scanf("%d", &ch);
                if(ch == 1)
                {
                    printf("Enter a node to insert to Front of SLL\n");
                    insert_front();
                }
                else if(ch == 2)
                {
                    delete_front();
                }
                else
                    printf("Invalid Entry\n");
                break;
        case 5: exit(0);
            default: printf("Enter the valid choice\n\n");
                 break;
            }
      }
      return 0;
}
```