Program:Write a menu driven program to implement SLL

```
//code to be executed
#include<stdlib.h>
#include <stdio.h>

void create();
void display();
void insert_begin();
void insert_end();
void delete_begin();
void delete_end();

struct node
{
        int info;
        struct node *next;
};
struct node *start=NULL;
int main()
{
```

```
        int choice;


                printf("\n  1.CREATE     ");
                printf("\n 2.DISPLAY  ");
                printf("\n 3. TO INSERT THE NODE AT THE
BEGINNING    ");
                printf("\n 4.TO INSERT THE NODE AT THE
END  ");
                printf("\n 5.TO DELETE THE NODE AT THE
BEGINNING      ");
                printf("\n 6.TO DELETE THE NODE AT THE
END        ");
                printf("\n 7.EXIT       ");


               while(1){
                  printf("\n enter the menu no. of your
choice:");
                  scanf("%d",&choice);


               switch(choice)
               {
                     case 1:
                                 create();
```

```
            break;
case 2:
            display();
            break;
case 3:
            insert_begin();
            break;
case 4:
            insert_end();
            break;
case 5:
            delete_begin();
            break;
case 6:
            delete_end();
            break;
case 7:
            exit(0);
            break;
default:
            printf("n Wrong Choice:n");
            break;
```

```
            }
        }
        return 0;
}
void create()
{
        struct node *temp,*ptr;
        temp=(struct node *)malloc(sizeof(struct
node));
        if(temp==NULL)
        {
                printf("Out of Memory Space:n");
                exit(0);
        }
        printf("Enter the value of the node which you
want to added:");
        scanf("%d",&temp->info);
        temp->next=NULL;
        if(start==NULL)
        {
                start=temp;
        }
```

```
        else
        {
                ptr=start;
                while(ptr->next!=NULL)
                {
                        ptr=ptr->next;
                }
                ptr->next=temp;
        }
}
void display()
{
        struct node *ptr;
        if(start==NULL)
        {
                printf("List is empty:n");
                return;
        }
        else
        {
                ptr=start;
                printf("The List elements are:");
```

```
            while(ptr!=NULL)
            {
                printf("%d\t",ptr->info );
                ptr=ptr->next ;
            }
        }
}
void insert_begin()
{
        struct node *temp;
        temp=(struct node *)malloc(sizeof(struct
node));
        if(temp==NULL)
        {
            printf("Out of Memory Space:n");
            return;
        }
        printf("Enter the value of the node for insert it
at beginning:" );
        scanf("%d",&temp->info);
        temp->next =NULL;
        if(start==NULL)
```

```
        {
                start=temp;
        }
        else
        {
                temp->next=start;
                start=temp;
        }
}
void insert_end()
{
        struct node *temp,*ptr;
        temp=(struct node *)malloc(sizeof(struct
node));
        if(temp==NULL)
        {
                printf("Out of Memory Space:n");
                return;
        }
        printf("Enter the value of node for insert it at
ending:" );
        scanf("%d",&temp->info );
```

```
        temp->next =NULL;
        if(start==NULL)
        {
                start=temp;
        }
        else
        {
                ptr=start;
                while(ptr->next !=NULL)
                {
                        ptr=ptr->next ;
                }
                ptr->next =temp;
        }
}
void delete_begin()
{
        struct node *ptr;
        if(ptr==NULL)
        {
                printf("List is Empty:n");
                return;
```

```
        }
        else
        {
                ptr=start;
                start=start->next ;
                printf("The deleted element from beginning
is :%d",ptr->info);
                free(ptr);
        }
}
void delete_end()
{
        struct node *temp,*ptr;
        if(start==NULL)
        {
                printf("List is Empty:");
                exit(0);
        }
        else if(start->next ==NULL)
        {
                ptr=start;
                start=NULL;
```

```
            printf("The deleted element from ending
is:%d",ptr->info);
            free(ptr);
      }
      else
      {
            ptr=start;
            while(ptr->next!=NULL)
            {
                  temp=ptr;
                  ptr=ptr->next;
            }
            temp->next=NULL;
            printf("The deleted element is:%d",ptr
->info);
            free(ptr);
      }
}


//output
```