```c
/* creation of singly linked list & Operations On Singly Linked List
    ->insertion
        ->at begning
        ->at end
        ->at specificposition
    -> Deletion
        ->at begning
        ->at end
        ->at specificposition
    ->find the length of list
    ->reverse the list
// Created By SMIT
*/
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>


//Global Declaractions
int count=0;
struct node{
    int data;
    struct node *next;
};
struct node *head=0,*temp;  //Global Pointers
void CreateList();  //To Create Link LIst
void InsAtBeg();    //To Insert Data At Beginig
void InsAtPos();    //This Function will Insert Data At Given Position
void InsAtEnd();    //To Insert Data At The End Of List
void DelAtBeg();    //To delete first Element/Node Of link list
void DelAtPos();    //To delete Spasific Node Of Link LIst
void DelAtEnd();    //To delete Last Node Of List
void FindLength();  //To find the length Of Link list
void ReverseList(); //To reverse The Link List
void Display();     //To Display Link LIst

int main(){
    int choice;
    while(choice!=11){      //This Loop Will Run Untill You Give 11 As an Input
        printf("\n**************MAIN MENU**************\n");
        //choice Must Beatween 1 To 11
        printf("1.Create a list\n");
        printf("2.insert at begning\n");
        printf("3.insert at  n(th) position\n");
        printf("4.insert at end\n");
        printf("5.delete from begning\n");
        printf("6.delete from n(th) position\n");
        printf("7.delete from end\n");
        printf("8.Find the length\n");
        printf("9.Reverse the Linked list\n");
        printf("10.Display Linked List\n");
        printf("11.exit\n");
        scanf("%d",&choice);

        //To Ran Spacific Function According To choice
    switch(choice){
        case 1:
            CreateList();
            break;
        case 2:
            InsAtBeg();
            break;
        case 3:
            InsAtPos();
            break;
        case 4:
            InsAtEnd();
            break;
        case 5:
            DelAtBeg();
            break;
        case 6:
            DelAtPos();
            break;
        case 7:
            DelAtEnd();
            break;
        case 8:
            FindLength();
            break;
        case 9:
            ReverseList();
            break;
        case 10:
            Display();
            break;
        case 11:
            exit(0);
            break;
        default:
            printf("\n\n\nEnter valid choice\n\n\n");
    }
}
    return 0;
// Created By SMIT
}

//This Function Will Create A Linked List
void CreateList(){
struct node *newnode;
int choice=1;
while(choice){
newnode=(struct node*)malloc(sizeof(struct node));
printf("Enter Data ");
scanf("%d",&newnode->data);
newnode->next=0;
if(head==0){
head=temp=newnode;
}
else{
    temp->next=newnode;
    temp=newnode;
}
printf("\nPress 0 to exit Press 1 To Add ");
scanf("%d",&choice);
count++;
}
printf("\n\n\n%d nodes created successfully\n\n\n",count);
}

//This Function Will Display The List
void Display(){
    if(head==0){
        printf("\n\n\nI Think You should Create A list First\n\n\n");
    }
    else{
        temp=head;
        printf("\n\n\nThe List is");
        while(temp!=0){
            printf("\n%d\n",temp->data);
            temp=temp->next;
        }
    }
}

//To Insert Node At Begning
void InsAtBeg(){
    if(head==0){
        printf("\n\n\nPlease Create a List First\n\n\n");
    }
    else{
    struct node *newnode;
    newnode=(struct node *)malloc(sizeof(struct node));
    printf("Enter data");
    scanf("%d",&newnode->data);
    newnode->next=head;
    head=newnode;
    }
}

//To Insert Node At spacific Positon
void InsAtPos(){
    if(head==0){
        printf("\n\n\nPlease Create a List First\n\n\n");
    }
    else{
        int pos,i=1;
        printf("Enter Position");
        scanf("%d",&pos);
        if(pos>count){
            printf("Invalid Position");
        }
        else{
            struct node *newnode;
            temp=head;
            while(i<pos){
                temp=temp->next;
                i++;
            }
            newnode=(struct node*)malloc(sizeof(struct node));
            printf("Enter Data");
            scanf("%d",&newnode->data);
            newnode->next=temp->next;
            temp->next=newnode;
        }
    }
}

//To Insert Node At The End Of List
void InsAtEnd(){
    if(head==0){
        printf("\n\n\nPlease Create a List First\n\n\n");
    }
    else{
        struct node *newnode;
        temp=head;
        while(temp->next!=0){
            temp=temp->next;
        }
        newnode=(struct node*)malloc(sizeof(struct node));
        printf("Enter Data");
        scanf("%d",&newnode->data);
        newnode->next=0;
        temp->next=newnode;
    }
}

//To Delete First Node
void DelAtBeg(){
    if(head==0){
        printf("\n\n\nPlease Create a List First\n\n\n");
    }
    else{
        temp=head;
        head=temp->next;
        free(temp);
        printf("\n\n\nDeleted successfully..\n\n\n");
    }
}
// Created By Smit
//To Delete Node AT Spacific Position
void DelAtPos(){
    int pos,i=1;
    if(head==0){
        printf("\n\n\nPlease Create A list First\n\n\n");
    }
    else{
        struct node *deletethis;
        temp=head;
        printf("Enter Position ");
        scanf("%d",&pos);
        if(pos>count){
            printf("\n\nInvalid Position\n\n");
        }
        else{
            while(i<pos-1){
                temp=temp->next;
                i++;
            }
            deletethis=temp->next;
            temp->next=deletethis->next;
            free(deletethis);
            printf("\n\n\nDeleted successfully...\n\n\n");
        }
    }
}

//To Delete Last Node
void DelAtEnd(){
    if(head==0){
        printf("\n\n\nPlease Create A List First\n\n\n");
    }
    else{
        struct node *prev;
        temp=head;
        while(temp->next!=0){
            prev=temp;
            temp=temp->next;
        }
        free(temp);
        prev->next=0;
        printf("\n\n\nDeleted Successfully...\n\n\n");
    }
}

//To Find The Lenth Of List
void FindLength(){
    if(head==0){
        printf("\n\n\nYou can't find length Without Creating List\n\n\n");
    }
    else{
    temp=head;
    count=0;
    while(temp!=0)  {
        temp=temp->next;
        count++;
    }
    printf("\n\n\nThe Length Of Linked List Is %d \n\n\n",count);
    }
}

//To Reverse The List
void ReverseList(){
    if(head==0){
        printf("\n\n\nThere Is No List To Reverse\n\n\n");
    }
```

```c
    else{
        struct node *previousnode,*currentnode,*nextnode;
        previousnode=0;
        currentnode=nextnode=head;
        while(nextnode!=0){
        nextnode=nextnode->next;
        currentnode->next=previousnode;
        previousnode=currentnode;
        currentnode=nextnode;
        }
        head=previousnode;
        printf("\n\n\nList Reversed Successfully..\n\n\n");
    }
}
// Created By SMIT

```