Start here | Lab1stack.c | Lab3aLinearqueue.c | Lab3bCircularqueue.c | Lab4SLL.c

```c
#include<stdio.h>
#include<stdlib.h>
struct Node
{
    int data;
    struct Node *next;
};

struct Node *createNode(int data)
{
    struct Node *newNode=(struct Node*)malloc(sizeof(struct Node));
    if(newNode==NULL)
    {
        printf("Memory allocation failed!\n");
        exit(1);
    }
    newNode->data=data;
    newNode->next=NULL;
    return  newNode;
};
void insertatBeginning(struct Node **head,int data)
{
    struct Node *newNode=createNode(data);
    newNode->next=*head;
    *head=newNode;
}

void insertatEnd(struct Node **head,int data)
{
    struct Node *newNode=createNode(data);
    if(*head==NULL)
    {
        *head=newNode;
        return ;
    }
    struct Node *temp=*head;
    while(temp->next!=NULL){
        temp=temp->next;
    }
    temp->next=newNode;

}
```

Start here | Lab1stack.c | Lab3aLinearqueue.c | Lab3bCircularqueue.c | Lab4SLL.c

```c

void insertatPosition(struct Node **head,int data,int position){
    if(position<1){
        printf("Invalid position !\n");
        return ;
    }
    if(position==1){
        insertatBeginning(head,data);
        return ;
    }
    struct Node *newNode=createNode(data);
    struct Node *temp=*head;
    for(int i=1;temp!=NULL&&i<position-1;i++)
        temp=temp->next;
    if(temp==NULL){
        printf("Position out of range!\n");
        free(newNode);
        return;
    }
    newNode->next=temp->next;
    temp->next=newNode;
}

void displaylist(struct Node *head){
    if(head==NULL){
        printf("List is empty.\n");
        return ;
    }
    printf("Linked List:");
    struct Node *temp=head;
    while(temp!=NULL){
        printf("%d->",temp->data);
        temp =temp->next;
    }
    printf("NULL\n");
}

int main()
{
    struct Node *head=NULL;
    int choice,data,position;
    while(1){
        printf("\n--Singly Linked list menu--\n");
```

Start here | Lab1stack.c | Lab3aLinearqueue.c | Lab3bCircularqueue.c | Lab4SLL.c

```c
    }
    printf("NULL\n");
}

int main()
{
    struct Node *head=NULL;
    int choice,data,position;
    while(1){
        printf("\n--Singly Linked list menu--\n");
        printf("1.Insert at Beginning\n");
        printf("2.Insert at End\n");
        printf("3.Insert at Position\n");
        printf("4.Display List\n");
        printf("5.Exit\n");
        printf("Enter your choice:");
        scanf("%d",&choice);
        switch(choice){
            case 1:printf("Enter Data:");
                   scanf("%d",&data);
                   insertatBeginning(&head,data);
                   break;
            case 2:printf("Enter data:");
                   scanf("%d",&data);
                   insertatEnd(&head,data);
                   break;
            case 3:printf("Enter data:");
                   scanf("%d",&data);
                   printf("Enter Position:");
                   scanf("%d",&position);
                   insertatPosition(&head,data,position);
                   break;
            case 4:displaylist(head);
                   break;
            case 5:printf("Exciting ....\n");
                   exit(0);
            default:printf("Invalid choice! Try again.\n");
        }
    }
    return 0;
}
```


D:\Coding\C\LAB\Lab4SLL.exe
--Singly Linked list menu--
1.Insert at Beginning
2.Insert at End
3.Insert at Position
4.Display List
5.Exit
Enter your choice:1
Enter Data:23
--Singly Linked list menu--
1.Insert at Beginning
2.Insert at End
3.Insert at Position
4.Display List
5.Exit
Enter your choice:2
Enter data:46
--Singly Linked list menu--
1.Insert at Beginning
2.Insert at End
3.Insert at Position
4.Display List
5.Exit
Enter your choice:2
Enter data:69
--Singly Linked list menu--
1.Insert at Beginning
2.Insert at End
3.Insert at Position
4.Display List
5.Exit
Enter your choice:4
Linked List:23->46->69->NULL
--Singly Linked list menu--
1.Insert at Beginning
2.Insert at End
3.Insert at Position
4.Display List
5.Exit
Enter your choice:5
Exciting ....
Process returned 0 (0x0) execution time : 22.860 s