ACADEMY OF TECHNOLOGY


NAME : PRIYADARSHAN GHOSH

COLLEGE ROLL NO: 72

UNIVERSITY ROLL NO: 16900319072

DEPARTMENT: ECE-1(Y)

SEMESTER: 3rd

PAPER CODE : ES-CS391


# ➢ Laboratory *Assignment #11*

**Q1.WRITE A MENU DRIVEN PROGRAM TO PERFORM FOLLOWING OPERATIONS / APPLICATION USING**

**FUNCTIONS:**

**A) CREATION OF SINGLY LINKED LIST**

**B) DISPLAY OF SINGLY LINKED LIST**

**C) SORTING**

**D) REVERSE A LINKED LIST**

**E) MERGING OF TWO LINKED LISTS AND SPLITTING A LINKED LIST**

## Ans:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
typedef struct link{
        int data;
        struct link *next;
}node;
void createSLL(node *l)
{
        char ch;
        while(1)
        {
                printf("\n Enter data:");
                scanf("%d",&l->data);
                printf("\n Another node(y/n)?");
                ch=getch();
                if(ch=='n'||ch=='N')
```

```
            {
                l->next=NULL;
                return ;
            }
            l->next=(node *)malloc(sizeof(node));
            l=l->next;
        }
}

void displaySLL(node *l)
{
        while(l->next!=NULL)
        {
            printf("\t %d",l->data);
            l=l->next;
        }
        printf("\t %d",l->data);
}
void bubblesort(node *h)
{
        node *l1,*l2,*l3=NULL;
        int t;
        for(l1=h;l1->next!=NULL;l1=l1->next)
        {
            for(l2=h;l2->next!=l3;l2=l2->next)
            {
                if(l2->data>l2->next->data)
```

```
                {
                        t=l2->data;
                        l2->data=l2->next->data;
                        l2->next->data=t;
                }
        }
        l3=l2;
    }
}

node *reverseSLL(node *h)
{
    node *pre,*cur;
    pre=NULL;
    cur=h;
    while(h!=NULL)
    {
        h=h->next;
        cur->next=pre;
        pre=cur;
        cur=h;
    }
    return (pre);
}
node *mergeSLL(node *h1,node *h2)
{
    node *l;
```

```
        l=h1;
        while(l->next!=NULL)
        {
                l=l->next;
        }
        l->next=h2;
}

int countnode(node *l)
{
        int count=0;
        while(l!=NULL)
        {
                l=l->next;
                count++;
        }
        return count;
}

node *splitSLL(node *h)
{
        node *h1,*l;
        l=h;
        int count,i=1,pos;
        printf("\n Enter position from which you want to split:");
        scanf("%d",&pos);
        count=countnode(h);
```

```
        if(pos<count)
        {
                while(i<pos)
                {
                        l=l->next;
                        i++;
                }
        }
        else
        printf("\n Wrong input");
        h1=l->next;
        l->next=NULL;
        printf("\n 1st list is:");
        displaySLL(h);
        printf("\n 2nd list is:");
        displaySLL(h1);
}
int main()
{
        node *h,*h2;
        int ch;
        h=(node *)malloc(sizeof(node));
        while(1)
        {
                printf("\n 1.Press 1 to create a Singly linked list");
                printf("\n 2.Press 2 to display a Singly linked list");
                printf("\n 3.Press 3 to sort a Singly linked list");
```

```
printf("\n 4.Press 4 to reverse a Singly linked list");
printf("\n 5.Press 5 to merge two lists");
printf("\n 6.Press 6 to split two lists");
printf("\n 7.Press 7 to exit");
printf("\n Enter choice:");
scanf("%d",&ch);
switch(ch)
{
        case 1:createSLL(h);
                break;
        case 2:displaySLL(h);
                break;
        case 3:bubblesort(h);
                displaySLL(h);
                break;
        case 4:h=reverseSLL(h);
                displaySLL(h);
                break;
        case 5:printf("\n Create another list to merge with this
list");
                h2=(node *)malloc(sizeof(node));
                createSLL(h2);
                printf("\n list is:");
                displaySLL(h2);
                mergeSLL(h,h2);
                printf("\n After merging,the list is:");
                displaySLL(h);
                break;
```

```
                case 6:    splitSLL(h);
                           break;
                case 7:exit(0);

            }
        }
        return 0;
}
```

## OUTPUT =>

```
1.Press 1 to create a Singly linked list
 2.Press 2 to display a Singly linked list
 3.Press 3 to sort a Singly linked list
 4.Press 4 to reverse a Singly linked list
 5.Press 5 to merge two lists
 6.Press 6 to split two lists
 7.Press 7 to exit
 Enter choice:1

 Enter data:56

 Another node(y/n)?
 Enter data:
89
```

*Another node(y/n)?*

*Enter data:12*

*Another node(y/n)?*

*Enter data:6*

*Another node(y/n)?*

*Enter data:9*

*Another node(y/n)?*

*Enter data:1*

*Another node(y/n)?*

*Enter data:5*

*Another node(y/n)?*

*Enter data:32*

*Another node(y/n)?*

*Enter data:21*

*Another node(y/n)?*

*1.Press 1 to create a Singly linked list*

*2.Press 2 to display a Singly linked list*

*3.Press 3 to sort a Singly linked list*

4.Press 4 to reverse a Singly linked list

5.Press 5 to merge two lists

6.Press 6 to split two lists

7.Press 7 to exit

Enter choice:

2

56 89 12 6 9 1 5 32 21

1.Press 1 to create a Singly linked list

2.Press 2 to display a Singly linked list

3.Press 3 to sort a Singly linked list

4.Press 4 to reverse a Singly linked list

5.Press 5 to merge two lists

6.Press 6 to split two lists

7.Press 7 to exit

Enter choice:3

1 5 6 9 12 21 32 56 89

1.Press 1 to create a Singly linked list

2.Press 2 to display a Singly linked list

3.Press 3 to sort a Singly linked list

4.Press 4 to reverse a Singly linked list

5.Press 5 to merge two lists

6.Press 6 to split two lists

7.Press 7 to exit

Enter choice:4

89 56 32 21 12 9 6 5 1

*1.Press 1 to create a Singly linked list*

*2.Press 2 to display a Singly linked list*

*3.Press 3 to sort a Singly linked list*

*4.Press 4 to reverse a Singly linked list*

*5.Press 5 to merge two lists*

*6.Press 6 to split two lists*

*7.Press 7 to exit*

*Enter choice:5*

*Create another list to merge with this list*

*Enter data:6*

*Another node(y/n)?*

*Enter data:7*

*Another node(y/n)?*

*Enter data:98*

*Another node(y/n)?*

*Enter data:12*

*Another node(y/n)?*

*Enter data:7*

*Another node(y/n)?*

list is: 6 7 98 12 7

After merging,the list is: 89 56 32 21 12 9 6 5 1 6 7 98 12 7

1.Press 1 to create a Singly linked list

2.Press 2 to display a Singly linked list

3.Press 3 to sort a Singly linked list

4.Press 4 to reverse a Singly linked list

5.Press 5 to merge two lists

6.Press 6 to split two lists

7.Press 7 to exit

Enter choice:2

89 56 32 21 12 9 6 5 1 6 7 98 12 7

1.Press 1 to create a Singly linked list

2.Press 2 to display a Singly linked list

3.Press 3 to sort a Singly linked list

4.Press 4 to reverse a Singly linked list

5.Press 5 to merge two lists

6.Press 6 to split two lists

7.Press 7 to exit

Enter choice:6

Enter position from which you want to split:7

1st list is: 89 56 32 21 12 9 6

2nd list is: 5 1 6 7 98 12 7

*1.Press 1 to create a Singly linked list*

*2.Press 2 to display a Singly linked list*

*3.Press 3 to sort a Singly linked list*

*4.Press 4 to reverse a Singly linked list*

*5.Press 5 to merge two lists*

*6.Press 6 to split two lists*

*7.Press 7 to exit*

*Enter choice:7*

---

*Process exited after 121.1 seconds with return value 0*

*Press any key to continue . . .*