Date __/__/__
Page _____
SPLASH

# Lab-8

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <malloc.h>
#include <stdlib.h>
struct node
{
    int item;
    struct node *next;
};
typedef struct node * Node;

Node getNode()
{
    Node x;
    x = (Node) malloc (sizeof( struct node));
    return x;
}
Node insert_front (Node first, int data)
{
    Node new_node;
    new_node = getNode();
    new_node -> item = data;
    new_node -> next = NULL;
    if (first == NULL)
    {
        return new_node;
    }
```

```
    new_node -> next = first;
    first = new_node;
    return first;
}
Node delete_end(Node first)
{
    Node prev, cur;
    if (first == NULL)
    {
        printf("List is Empty and
                cannot be deleted\n");
        return first;
    }
    cur = first;
    while (cur -> next != NULL)
    {
        prev = cur;
        cur = cur -> next;
    }
    prev -> next = NULL;
    free(cur);
    return first;
}
void display(Node first)
{
    int count=0;
    Node temp;
    if (first == NULL)
    {
        printf("\n List is Empty\n");
    }
    for (temp = first; temp != NULL; temp
                                = temp -> next)
```

```
{
    count++;
    printf("%d\n", temp->item);
}
printf("\n Number of Nodes
        in the list are
        %d\n", count);
}
void search(Node first, int data)
{
    int pos=0
    Node temp;
    int i;
    if(first == NULL)
    {
    printf("List is empty\n");
    return;
    }
for(temp=first, i=0; temp!=NULL;
    temp=temp->next; i++)
{
    if(temp->item == data)
    {
    pos=i+1;
    printf("\n Search Successfull\n");
    printf("Element is found at
            position %d\n", pos);
    break;
    }
else
{
    pos=0;
```

```
        }
    }
    if(pos==0)
    {
        printf("\n Search Unsuccessfull \n");
    }
}
void sort(Node first)
{
    int t;
    Node temp;
    if(first==NULL)
    {
        printf("List is Empty \n");
        return;
    }
    for(Node i=first; i!=NULL; i=i->Next)
    {
        for(Node j=i->next; j!=NULL; j=j->Next)
        {
            if((i->item)>(j->item))
            {
                t=i->item;
                i->item=j->item;
                j->item=t;
            }
        }
    }
    printf("\n List is ascending order \n");
```

```
}
int main()
{
    Node first = NULL;
    Node a = NULL;
    Node b = NULL;
    Node ans = NULL;
    int choice, val, pos, n;
    do
    {
        printf("**** Actions ****\n");
        printf("1. Insert front\n");
        printf("2. Delete end\n");
        printf("3: Sort the list\n");
        printf("4. Search for an element\n");
        printf("5. Display the count
                and list\n");
        printf("\n Which action do yo
                wanna perform ?\n");
        scanf("%d", &choice);
        switch(choice)
        {
        case 1:
            printf("Enter the value to be
                    inserted\n");
            scanf("%d", &val);
            first = insert_front(first, val);
            break;
        case 2:
            first = delete_end(first);
            break;
```

```
case 3:
    sort (first)
    display (first);
    break;
case 4:
    printf ("Enter the number to
             Searched \n");
    scanf ("%d", &n);
    Search (first, n);
    break;
case 5:
    printf ("\n The elements are. \n");
    display (first);
    break;
}
while (choice != 6);
}
```