1) WAP

Implement Single Link List with following operations

a) Sort the linked list.

b) Reverse the linked list.

c) Concatenation of two linked lists

```
#include<stdio.h>

#include<stdlib.h>

struct Node

{

   int data;

   struct Node *link;

};

typedef struct Node node;

node *l1=NULL;

node *l2=NULL,*l3;

void display(node *start)

{

   node *temp;

   if(start==NULL)

   {

      printf("Linked list is empty\n");

      return;

   }

   temp=start;

   while(temp!=NULL)

   {

      printf("%d\n",temp->data);

      temp = temp->link;
```

```
    }
}
node* create()
{
    int c;
    node *new,*curr,*start=NULL;
    start=(node *) malloc(sizeof(node));
    curr=start;
    printf("Enter element\n");
    scanf("%d",&start->data);
    while(1)
    {
        printf("Do you want to add another element(1 for Yes / 0 for No)\n");
        scanf("%d",&c);
        if(c==1)
        {
            new=(node *) malloc(sizeof(node));
            printf("Enter element\n");
            scanf("%d",&new->data);
            curr->link = new;
            curr=new;
        }
        else
        {
            curr->link=NULL;
            break;
        }
```

```
    }
    return(start);
}
node* concat(node *start1,node *start2)
{
    node *start3,*temp;
    if(start2==NULL)
    {
        start2=start3;
        return(start3);
    }
    if(start1==NULL)
    {
        start3=start2;
        return(start3);
    }
    temp=start1;
    while(temp->link!=NULL)
    {
        temp=temp->link;
    }
    temp->link=start2;
    start3=start1;
    return(start3);
}

void reverse(node *start)
```

```
{


    node *a=start,*b=NULL,*c=NULL;
    while(a!=NULL)
    {
        c=b;
        b=a;
        a=a->link;
        b->link=c;
    }
    start=b;
    printf("Reversed linked list is:\n");
    display(start);
}
void sort(node *start)
{
    int count=0,t;
    node *a,*b,*temp;
    temp=start;
    while(temp!=NULL)
    {
        count++;
        temp=temp->link;
    }
    int n=count;
    a=start;
    b=start->link;
```

```
    for(int i=0;i<n-1;i++)
    {
        for(int j=0;j<n-i-1;j++)
        {
            if(a->data>b->data)
            {
                t=a->data;
                a->data=b->data;
                b->data=t;
            }
            a=b;
            b=b->link;
        }
        a=start;
        b=start->link;
    }
    printf("Sorted linked list is:\n");
    display(a);
}

void main()
{
    int ch;
    while(1)
    {
        printf("1.Merge 2.Reverse 3.Sort 4.Exit\n");
        printf("Enter your choice:\n");
```

```
    scanf("%d",&ch);
    switch(ch)
    {
    case 1:
       printf("Enter 1st linked list elements\n");
       l1=create();
       printf("Enter 2nd linked list elements\n");
       l2=create();
       l3=concat(l1,l2);
       printf("Merged linked list is:\n");
       display(l3);
       break;
    case 2:
       l1=create();
       reverse(l1);
       break;

    case 3:
       l1=create();
       sort(l1);
       break;
    case 4:
       exit(1);
    default:
       printf("Invalid choice\n");
    }
}}
```