LAB-7.

NAME: ANSHUL H. SURANA
USN: 1BM19CS020

1) WAP Implement Single Link list with following operations a) Sort the linkedlist b) Reverse the linked list c) Concatenation of two linked lists.

```
#include <stdio.h>
#include <stdlib.h>
void sort()
void create();
void reverse();
void create_second();
void concatenate();
void display();
struct node
{ int data;
  struct node *next;
};
struct node *head=NULL;
struct node *head2=NULL;
int c;
int main()
{   int choice;
    do
{ printf("\n1.Create \n2.Sort \n3.Reverse \n4.
  Enter second list \n5.Concatenation \n6.Display
  \n7.Exit");
  scanf("%d", &choice);
switch(choice)
{ case 1: create();
          break;
```

Anshul H. Surana

```
        case 2: sort();
                break;
        case 3: reverse();
                break;
        case 4: create_second();
                break;
        case 5: concatenate();
                break;
        case 6: display();
                break;
        case 7: exit(0);
    }
}while (choice!=7);
    return 0;
}

void create()
{   struct node *newnode;
    struct node *temp;
    int s;
    printf("Enter integer");
    scanf("%d", &s);
    newnode=(struct node*)malloc(sizeof(struct node
    newnode->data=s;
    if(head == NULL)
{   newnode->next=NULL;
    head=newnode;
    printf("First node created\n");
    ctf;
}else
    temp=head;
    while(temp->next!=NULL)
{   temp=temp->next;}}
```


Anshul H. Surana

NAME: ANSHUL H. SUPANA
USN: 1BM19CS020

```
        temp -> next = newnode;
        newnode -> next = NULL;
        ctt;
        printf("Node Created\n");
    }
}

void reverse()
{ struct node * prev=NULL, * current = head, * next=NULL;
  while (current != NULL)
  {   next = current -> next;
      current -> next = prev;
      prev = current;
      current = next;
  }
  head = prev;
  printf("The list is reversed\n");
}

void display()
{ struct node * ptr = NULL;
   ptr = head;
   if (ptr == NULL)
  { printf("List is empty\n");
  } else
   { printf("\n contents of the linked list");
     while (ptr != NULL)
     {
      printf("\t %d", ptr -> data);
       ptr = ptr -> next;
     }
}
printf("\n");
}
```

Anshul H Supana

NAME: ANSHUL H SUPANA
USN: 1BM19CS020

```
void create_second()
{ struct node * newnode;
  struct node * temp;
    int s,y;

  printf("Enter integer:");
  scanf("%d",&s);
  newnode=(struct node*)malloc(size of (struct node
  newnode->data=s;
    if(head2==NULL)
{ newnode->next=NULL;
    head2=newnode;
    printf("First node created\n");
    ct++;
} else
  { temp=head2;
    while(temp->next!=NULL)
    {    temp=temp->next;  }
    temp->next=newnode;
    newnode->next=NULL;
    ct++;
    printf("Node created\n");
    }
}
void concatenate()
{   struct node *ptr;
    if(head==NULL)
    { head2=head; }
    ptr=head;
  while(ptr->next!=NULL)
    ptr=Ptr->next;
```


Anshul H. Supana

NAME: ANSHUL H. SURANA
USN: 1BM19CS020

```
        ptr -> next = head2;
        printf("The list is concatenated\n");
}
void sort()
{
        int swap, i;
        struct node *ptr1;
        struct node *lptr = NULL;

        if (head == NULL)
                return;
do {
        swap = 0;
        ptr1 = head;
        while (ptr1 -> next != lptr)
{       if (ptr1 -> data > ptr1 -> next -> data)
        {
                int temp = ptr1 -> data;
                ptr1 -> data = ptr1 -> next -> data;
                ptr1 -> next -> data = temp;
                swap = 1;
        }
        ptr1 = ptr1 -> next;
} lptr = ptr1;
}
        while (swap);
        printf ("The list is sorted\n");
}
```

Anshul H. Surana