LAB PROGRAM-7

1BM19CS052

```
# include <stdio.h>
# include <stdlib.h>

struct node
{
    int data;
    struct node * next;
};

void insertBack ( struct node ** headptr, int
                                        value )
{
    struct node * newnode, * temp.
    newnode = (struct node *) malloc (sizeof
                    (struct node));
    newnode -> data = value;
    newnode -> next = NULL;
    temp = * headptr;
    if (temp == NULL)
    {
        * headptr = newnode;
    }
    else
    {
        while (temp -> next != NULL)
            temp = temp -> next;
        temp -> next = newnode;
    }
}
```

```
void removeBack( struct node ** headptr
{
   struct node * temp;
   temp = * headptr;
   if ( temp = NULL)
   {
      printf ("The list is Empty !!! \n");
      return;
   }
  else.
  {
     while ((temp->next)->next != NULL)
         temp = temp->next;
     temp->next = NULL;
     printf ("Last element has been. Deleted\n");
  }
}
  void display( struct node * temp)
  {
    if (temp ==NULL)
    {
       printf ("The list is Empty \n");
       return;
    }
   else
   {
       while (temp!=NULL)
       {
          printf ("%d\t", temp->data);
          temp = temp->next;
       }
    printf ("\n");
  }
}

void sort (struct node **headptr)
{
   struct node *p, *q;
    p = *headptr;
    int temp;
    if (p==NULL)
    {
      printf ("List is Empty!\n");
      return;
    }
    for(; p!=NULL; p=p->next)
    {
      for(q=p->next; q!=NULL; q=q->next)
      {
        if (p->data > q->data)
        {
          temp = q->data;
          q->data = p->data;
          p->data = temp;
        }
      }
    }
    printf ("Sort Complete!\n");
}
  void reverse (struct node *temp)
{
  if (temp == NULL)
```

```
{
  printf ("List is empty \n");
  return;
}
struct node *first = NULL, *second = temp,
   *third = NULL;
while (second != NULL)
{
    third = second -> next;
    second -> next = first;
    first = second;
    second = third;
}
    temp = first;
    printf ("After reversal:\n");
    while (temp != NULL)
    {
      printf ("%d\t", temp -> data);
       temp = temp => ~~tc~~ next;
    }
    printf ("\n");
}
  void ~~(co~~ concatenate (struct node *temp1,
       struct node *temp2)
 {
    if (temp1 == NULL && temp2 == NULL)
   {
      printf ("Both lists are empty \n");
    }
     else if (temp2 == NULL && temp1 != NULL)
     {
```

```
printf (" After Concatenation: \n");
while (temp1 != NULL)
{
    printf ("%d\t", temp1->data);
    temp1 = temp1->next;
}
    printf ("\n");
```