LAB PROGRAM 7

```
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 (" 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");
              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 completed!!! \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 -> next ;
   }
   printf ("\n");
}
void    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" , temp-> data);
                temp1  =  temp1 -> next;
           }
          printf("\n");
      }
      else  if ( temp1  == NULL  && temp2 != NULL)
      {
           printf (" After concatenation:\n") ;
           while      ( temp2 != NULL)
           {
                printf ("%d\t", temp2 -> data);
                temp2  = temp2 -> next;
            }
            printf ("\n");
       }
       else
      {      struct     node * ry = temp1;
             while  ( temp1->next != NULL)
                    temp1  =  temp1 -> next;
             temp1 -> next = temp2;
             printf (" After   concatenation: \n");
             temp1 = ry;
while (temp1 != NULL)
{       printf ("%d \t", temp1 -> data);
         temp1 = temp1 -> next;
 }
    printf (" \n");
  }
}
```