1.//program for the Insertion sort algorithm//

```
#include<stdio.h>
int main()
{
    int i, m, ele_n,temp, val[25];
    printf("Enter the number of elements : ");
    scanf("%d",&ele_n);
    printf("Enter elements: ");
    for(i=0;i<ele_n;i++)
    scanf("%d",&val[i]);
    for(i=1;i<ele_n;i++)
    {
        temp=val[i];
        m=i-1;
        while((temp<val[m])&&(m>=0))
        {
            val[m+1]=val[m];
            m=m-1;
        }
        val[m+1]=temp;
    }
    printf("Sorted elements in ascending order : ");
    for(i=0;i<ele_n;i++)
    printf(" %d\t",val[i]);
    return 0;
}
```

2.//program for the Selection sort algorithm//

```
#include<stdio.h>
int main()
{
   int i, j, num, temp, val[25];
   printf("Enter the number of elements : ");
   scanf("%d",&num);
   printf("Enter elements: ");
   for(i=0;i<num;i++)
   scanf("%d",&val[i]);
   for(i=0;i<num;i++)
   {
      for(j=i+1;j<num;j++)
      {
```

```
        if(val[i]>val[j])
        {
          temp=val[i];
          val[i]=val[j];
          val[j]=temp;
        }
      }
   }
   printf("Sorted elements in ascending order: ");
   for(i=0;i<num;i++)
     printf(" %d",val[i]);
   return 0;
}
```

3.// program for bubble sort algorithm //

```
#include<stdio.h>
int main()
{
    int count, temp, i, h, val[30];
    printf("Enter the number of elements : ");
    scanf("%d",&count);
    printf("Enter %d numbers: ",count);

    for(i=0;i<count;i++)
    scanf("%d",&val[i]);
    for(i=count-2;i>=0;i--)
    {
       for(h=0;h<=i;h++)
       {
          if(val[h]>val[h+1])
          {
             temp=val[h];
             val[h]=val[h+1];
             val[h+1]=temp;
          }
       }
    }

   printf("Sorted elements : ");
   for(i=0;i<count;i++)
     printf(" %d",val[i]);

   return 0;
```

```
}

4.// program for Merge Sort //
#include<stdlib.h>
#include<stdio.h>
void merge(int arr[], int l, int m, int r)
{
    int i, j, k;
    int n1 = m - l + 1;
    int n2 =  r - m;

    /* create temp arrays */
    int L[n1], R[n2];

    /* Copy data to temp arrays L[] and R[] */
    for (i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[m + 1+ j];

    /* Merge the temp arrays back into array*/
    i = 0; // Initial index of first subarray
    j = 0; // Initial index of second subarray
    k = l; // Initial index of merged subarray
    while (i < n1 && j < n2)
    {
        if (L[i] <= R[j])
        {
            arr[k] = L[i];
            i++;
        }
        else
        {
            arr[k] = R[j];
            j++;
        }
        k++;
    }

    /* Copy the remaining elements of L[] if any*/
    while (i < n1)
    {
        arr[k] = L[i];
```

```
        i++;
        k++;
    }

    /* Copy the remaining elements of R[], if any */
    while (j < n2)
    {
        arr[k] = R[j];
        j++;
        k++;
    }
}
void mergeSort(int arr[], int l, int r)
{
    if (l < r)
    {
        int m = l+(r-l)/2;

        // Sort first and second halves
        mergeSort(arr, l, m);
        mergeSort(arr, m+1, r);

        merge(arr, l, m, r);
    }
}

/* Function to print an array */
void printArray(int A[], int size)
{
    int i;
    for (i=0; i < size; i++)
        printf("%d ", A[i]);
    printf("\n");
}

int main()
{
    int siz,v;
    printf("Enter array size : ");
    scanf("%d",&siz);
    int val[siz];
    for(v=0;v<siz;v++)
    {
```

```
        printf("Enter Value :");
        scanf("%d",&val[v]);
    }
    printf("Given array is \n");
    printArray(val,siz);
    mergeSort(val, 0, siz-1);

    printf("\nSorted array is \n");
    printArray(val,siz);
```

-

M R M Sitarama Reddy
AP19110010363
CSE-F