1.Write a program for the Insertion sort algorithm.

```
//program for the Insertion sort algorithm//
#include<stdio.h>
int main(){

  int c, h, count_num, temp, numericals[25];

  printf("Enter the number of elements you are going to enter?: ");
  scanf("%d",&count_num);
  printf("Enter %d elements: ", count_num);
  for(c=0;c<count_num;c++)
    scanf("%d",&numericals[c]);
  for(c=1;c<count_num;c++){
    temp=numericals[c];
    h=c-1;
    while((temp<numericals[h])&&(h>=0)){
      numericals[h+1]=numericals[h];
      h=h-1;
    }
    numericals[h+1]=temp;
  }
  printf("Sorted elements according to ascending order: ");
  for(c=0;c<count_num;c++)
    printf(" %d",numericals[c]);
  return 0;
}
```

---

2. Write a program for the Selection sort algorithm.

```
//program for the Selection sort algorithm//

#include<stdio.h>
int main(){
```

```
  int c, h, count_num, temp, numericals[25];
  printf("Enter the number of elements you are going to enter?: ");
  scanf("%d",&count_num);
  printf("Enter %d elements: ", count_num);
  for(c=0;c<count_num;c++)
    scanf("%d",&numericals[c]);
  for(c=0;c<count_num;c++){
    for(h=c+1;h<count_num;h++){
      if(numericals[c]>numericals[h]){
        temp=numericals[c];
        numericals[c]=numericals[h];
        numericals[h]=temp;
      }
    }
  }
  printf("Here prints the Sorted elements according to your input: ");
  for(c=0;c<count_num;c++)
    printf(" %d",numericals[c]);
  return 0;
}
```

---

3. Write a program for Bubble sort algorithm.

```
// program for bubble sort algorithm //
#include<stdio.h>
int main(){

  int count_num, temp, c, h, numericals[30];

  printf("Enter the number of elements you are going to enter?: ");
  scanf("%d",&count_num);

  printf("Enter %d numericals: ",count_num);
```

```
  for(c=0;c<count_num;c++)
  scanf("%d",&numericals[c]);
  for(c=count_num-2;c>=0;c--){
    for(h=0;h<=c;h++){
      if(numericals[h]>numericals[h+1]){
        temp=numericals[h];
        numericals[h]=numericals[h+1];
        numericals[h+1]=temp;
      }
    }
  }

  printf("Sorted elements according to ascending order:");
  for(c=0;c<count_num;c++)
    printf(" %d",numericals[c]);

  return 0;
}
```

4.Write a program for the Merge sort algorithm.

```
// program for Merge Sort //
#include<stdlib.h>
#include<stdio.h>

void merge(int array[], int c, int h, int a)
{
            int i, j, d;
            int num1 = h - c + 1;
            int num2 = a - h;
            int C[num1], A[num2];
            for (i = 0; i < num1; i++)
```

```
        C[i] = array[c + i];
for (j = 0; j < num2; j++)
        A[j] = array[h + 1+ j];
i = 0;
j = 0;
d = c;
while (i < num1 && j < num2)
{
        if (C[i] <= A[j])
        {
                array[d] = C[i];
                i++;
        }
        else
        {
                array[d] = A[j];
                j++;
        }
        d++;
}
while (i < num1)
{
        array[d] = C[i];
        i++;
        d++;
}
while (j < num2)
{
        array[d] = A[j];
        j++;
        d++;
```

```
        }
}
void mergeSort(int array[], int c, int a)
{
        if (c < a)
        {
                int h = c+(a-c)/2;
                mergeSort(array, c, h);
                mergeSort(array, h+1, a);
                merge(array, c, h, a);
        }
}
void printarrayay(int A[], int size)
{
        int i;
        for (i=0; i < size; i++)
                printf("%d ", A[i]);
        printf("\n");
}
int main()
{
        int array[] = {1, 9, 1, 1, 0, 0, 1, 0, 5, 3, 6};
        int array_size = sizeof(array)/sizeof(array[0]);
        mergeSort(array, 0, array_size - 1);
        printf("\nSorted arrayay is in ascending order \n");
        printarrayay(array, array_size);
        return 0;
}
```

**********THE-END**********