K.VARSHINI REDDY
AP19110010403
CSE-F

1. **Write a program for the Insertion sort algorithm.**

CODE:

```
#include<stdio.h>
int main(){
 int i, j, count, temp, number[10];
 printf("Enter the number of elements ");
 scanf("%d",&count);
 printf("Enter %d elements: ", count);
 for(i=0;i<count;i++)
 scanf("%d",&number[i]);
 for(i=1;i<count;i++){
 temp=number[i];
 j=i-1;
 while((temp<number[j])&&(j>=0)){
 number[j+1]=number[j];
 j=j-1;
 }
 number[j+1]=temp;
 }
 printf("Order of Sorted elements: ");
 for(i=0;i<count;i++)
 printf(" %d",number[i]);
 return 0;
}
```

2. **Write a program for the Selection sort algorithm.**

CODE:

```
#include<stdio.h>
int main(){
 int i, j, count, temp, number[25];
```

```
 printf("Enter the number of elements");
  scanf("%d",&count);
 printf("Enter %d elements: ", count);
 for(i=0;i<count;i++)
 scanf("%d",&number[i]);
 for(i=0;i<count;i++){
 for(j=i+1;j<count;j++){
 if(number[i]>number[j]){
 temp=number[i];
 number[i]=number[j];
 number[j]=temp;
 }
 }
 }
 printf("Sorted elements: ");
 for(i=0;i<count;i++)
 printf(" %d",number[i]);
 return 0;
}
```

3.Write a program for the Bubble sort algorithm.

CODE:

```
#include<stdio.h>
int main(){
 int count, temp, i, j, number[30];
 printf("Enter the number of elements");
 scanf("%d",&count);
 printf("Enter %d numbers: ",count);
 for(i=0;i<count;i++)
 scanf("%d",&number[i]);
 for(i=count-2;i>=0;i--){
 for(j=0;j<=i;j++){
 if(number[j]>number[j+1]){
 temp=number[j];
 number[j]=number[j+1];
 number[j+1]=temp;
 }
```

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

4. Write a program for the Merge sort algorithm.

CODE:

```
#include<stdio.h>
#include<conio.h>
#define MAX_SIZE 10
void merge_sort(int, int);
void merge_array(int, int, int, int);
int arr_sort[MAX_SIZE];
int main() {
 int i;
 printf("\nEnter %d Elements for Sorting\n", MAX_SIZE);
 for (i = 0; i < MAX_SIZE; i++)
 scanf("%d", &arr_sort[i]);
 printf("\nYour Data :");
 for (i = 0; i < MAX_SIZE; i++) {
 printf("\t%d", arr_sort[i]);
 }
 merge_sort(0, MAX_SIZE - 1);
 printf("\n\nSorted Data :");
 for (i = 0; i < MAX_SIZE; i++) {
 printf("\t%d", arr_sort[i]);
 }
 getch();
}
void merge_sort(int i, int j) {
 int m;
 if (i < j) {
```

```
 m = (i + j) / 2;
 merge_sort(i, m);
 merge_sort(m + 1, j);
 merge_array(i, m, m + 1, j);
 }
}
void merge_array(int a, int b, int c, int d) {
 int t[50];
 int i = a, j = c, k = 0;
 while (i <= b && j <= d) {
 if (arr_sort[i] < arr_sort[j])
 t[k++] = arr_sort[i++];
 else
 t[k++] = arr_sort[j++];
 }
 while (i <= b)
 t[k++] = arr_sort[i++];
 while (j <= d)
 t[k++] = arr_sort[j++];
 for (i = a, j = 0; i <= d; i++, j++)
 arr_sort[i] = t[j];
}
```