Lab programs

S. Tulasi
AP19110010452
CSE-F

1) Insertion sort :-

```
#include <stdio.h>
int main()
{
    int n, i, j, temp;
    int A[20];
    printf("Enter the number of elements");
    scanf("%d",&n);
    for(i=0; i<n; i++){
        printf("Enter element\n");
        scanf("%d", &A[i]);
  }
  for(i=1; i<=n-1; i++) {
        j=i;
        while (j>0 && A[j-1]>A[j]){
            temp= A[j];
            A[j] = A[j-1];
            A[j] = temp;
        }   j--;
}
Printf("SORTED ARRAY");
for(i=0; i<n; i++){
    printf("%d \t", A[i]);
}
return 0;
}
```

2) Selection sort :-

```
#include <stdio.h>
int main(){
    int i, j, temp, n;
    int A[20];
    printf("Enter the number of elements");
    scanf("%d", &n);
    for(i=0; i<n; i++){
      printf("Enter the element");
      scanf("%d", &A[i]);
    }
    for(i=0; i<n; i++){
       for(j=i+1; j<n; j++){
          if(A[j] < A[i]){
              temp = A[i];
              A[i] = A[j];
              A[j] = temp;
          }
    }  }
    printf("SORTED ARRAY");
    for(i=0; i<n; i++){
      printf("%d\t", A[i]);
    }
    return 0;
}
```

3) Bubble sort :-

```
#include <stdio.h>
int main()
{
   int i, j, temp, n;
   int A[20];
   printf("Enter the number of elements");
   scanf("%d", &n);
   for(i=0; i<n; i++){
     printf("Enter the element");
     scanf("%d", &A[i]);
   }
    for(i=0; i<n; i++){
       for(j=0; j<n-i-1; j++){
          if(A[j] > A[j+1]){
             temp= A[j];
             A[j]= A[j+1];
              A[j+1]= A[j];
          }
    }
}
printf("SORTED ARRAY");
for(i=0; i<n; i++){
   printf("%d", A[i]);
}
return 0;
```

4) Merge sort :-

```
#include <stdio.h>
void merge (int A[], int i1, int j1, int i2, int j2);
void mergesort (int A[], int i, int j);
int main () {
    int A[30], n, i;
    printf ("Enter number of elements");
    scanf ("%d", &n);
    for(i=0; i<n; i++) {
        printf("Enter the element");
        scanf ("%d", &A[i]);
        mergesort (A, 0, n-1);
        printf("SORTED ARRAY\n");
        for(i=0; i<n; i++) {
            printf ("%d\t", A[i]);
        }
        return 0;
}
void mergesort (int A[], int i, int j) {
        int mid;
        if (i<j) {
            mid = (i+j)/2;
            mergesort(A, i, mid);
            mergesort(A, mid+1, j);
            merge(A, i, mid, mid+1, j);
        }
}
```

```
void merge (int A[], int i1, int j1, int i2, int j2) {
    int temp[50];
    int i, j, k;
    i = i1;
    j = i2;
    k = 0;
    while(i<j1 && j<=j2) {
        if (A[i]<A[j]){
            temp[k++] = A[i++];
        }
        else {
            temp[k++] = A[j++];
        }
        while (i<j1){
            temp[k++] = A[i++];
        }
        while (j<j2) {
            temp [k++] = A[j++];
        }
        for(i=i1, j=0; i<=j2; i++, j++) {
            A[i] = temp[j];
}
```