Lab programs

M. Jaswanth
AP19110010429
CSE-H

1) write a program for insert sort algorithm.

Ans)

```
#include<stdio.h>
void main().
{
   int n, array[100], c, d, t;
   printf("Enter number of elements \n");
   scanf("%d", &n);
   printf("Enter %d integers \n", n);
   for (c=0; c<n; c++){
         scanf("%d", &array[c]);
   }
   for (c=1; c<=n-1; c++){
         d=c;
         while(d>0 && array[d-1]>array[d]){
                t=array[d];
                array[d]=array[d-1];
                array[d-1]=t;
                d--;
         }
   }
   printf("Sorted array in ascending order: \n");
   for (c=0; c<=n-1; c++){
         printf("%d \n", array[c])
   }.
```

2) Write a program for the selection sort.

A)

```
#include<stdio.h>
void main()
{
  int array[1000], n, c, d, position, temp;
  printf("Enter number of elements: \n");
  scanf("%d", &n);
  printf("Enter %d integer \n", n);
  for(c=0; c<n; c++){
        scanf("%d", &array[c]);
  }
  for(c=0; c<<n-1); c++){
        position=c;
        for(d=c+1; d<n; d++){
            if (array[position]>array[d])
                position=d;
            }
            if (position!=c){
                temp=array[c];
                array[c]=array[position];
                array[position]=temp;
            }
        }
        printf("Sorted array in ascending order: \n");
        for (c=0; c<n; c++){
            printf("%d\n", array[c]);
        }
    }
}
```

3) write a program for bubble sort algorithm:

4)

```
#include<stdio.h>
void main()
{
  int array[1000], n, c, d, position, temp;
  printf("Enter number of element \n");
  scanf("%d", &n);
  printf("Enter %d integers \n", n);
  for (c=0; c<n; c++){
      scanf("%d", &array[c]);
  }
  for (c=0; c<(n-1); c++) {
     for(d=0; d<n-c-1; d++){
        if (array[d]>array[d+1]){
            temp=array[d];
            array[d+1]=array[d+1]
            array[d+1] = temp;
        }.
     }
  }.
  printf("Sorted list in ascend order: \n");
  for(c=0; c<n; c++){
     printf("%d\n", array[c]);
  }.
}.
```

4) write a program for the merge sort algorithm.

A)

```
#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;
  int L[n1], R[n2];
  for (i=0; i<n1; i++)
      L[i]=arr[l+i];
  for (j=0; j<n2; j++)
      R[j]=arr[m+l+j];
    i=0;
    j=0;
    k=l;
  while (i<n2 && j<n2).
  {
    if (L[i]<=R[j]).
    {
      arr[k]=L[i];
        i++;
    } }.
    else
    {
      arr[k]=R[j].
        j++;
    }.   k++;
    }k++;
```

```
while (i<n1)
{
    arr[k] = L[i];
    i++;
    k++;
}

while(j<n2)
{
    arr[k] = R[j];
    j++;
    k++;
}
}

void mergesort(int arr[], int l, int r)
{
if (l<r)
{
int m = l+(r-1)/2
mergesort(arr, l, m);
mergesort(arr, m+1, r);
merge Sort(arr, l, m, r);
}
}

void printArray(int A[], int size)
{
int i;
for(i=0; i<size; i++)
printf("%d ", A[i]);
printf("\n");
}
```

```
int main()
{
  int arr[]2 { 9, 10, 15, 8, 4};
  int arr_size2 size or(arr)/size of (arr[0]);
  printf("Given array is \n");
   printf 'Array (arr, arr_size);
   mergeSort(arr, 0, arr_size-1).
   printf("\n sorted array is \n");
   printArray (arrg arr_size);
    return 0;
}
```