DSA
Lab programs on SORTING

SRIKAR SRIPADA
AP19110010474
CSE-F

(1)

Insertion sort :-

```
#include < stdlib.h>
#  include <stdio.h>
   Void main()
{
   int n, array[100], C, S, k;
   Printf ("Enter no of elements \n");
   Scanf ("%d", &n);
   Printf("Enter %d integers\n", n);
   for (c=0; c<=n-1; c++)
   {
    s=c ;
   while (d>0 && array[s-1] >array[s]
   {
        K = array [s];
        array[s] = array [s-1];
        array [s-1] = k;
          s--;
     }
}
   Printf ("sorted array in A.O ie ascendingorder :\n");
   for (c=0; c<=n-1; c++)
    {
      Printf(" %d\n", array [c]);
     }
```

Scanned with CamScanner

Input:-

Enter no of elements in array:

```
5
```

Enter elements:

```
18
3
2
5
```

Out Put:-

Sorted array in A-O ie assending order:

```
2
3
5
18.
```

(2) Selection sort:-

```
# include <stdio.h>
  voidmain()
{
  int  array[100], c ,s, k, Position temp ;
  Print f (" Enter no 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 (s = c+1; s<n; s++)
{
 if (array [Position] > array [s])
     Position = s;
}
if (Position! = c)
{
temp = array [c];
array [c] = array [Position];
array [Position] = temp;
}
}
  Printf ("array in A.O : \n");
  for (c = 0; c<n; c++)
  {
      Printf("%d \n", array [c]);
  }
}
```

```
Out Put / Input :-
  Enter no. of elements
  5
Enter
  3
  4
  1
  2
  3
```

```
array in A.O :
   1
   2
   3
   4
   5
```

③ Bubble Sort :-

```
# include<stdlib.h>
# include<stdio.h>
void main()
{
   int  array[100], c, s, n, position, temp;
   Printf ("Enter no. of Elements\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 ( s=0; s<n-c-1; s++)
       {
        if (array[s] > array[s+1])
         {
           temp = array[s];
           array[s] = array[s+1];
          array[s+1] = temp;

         }
      }
              Printf ("Sorted list in A.O :\n");
          for (c=0; c<n ; c++)
           {
          Printf("%d\n", array[c]);
           }
        }
```

Scanned with CamScanner

InPut and out Put :-

```
Enter no. of Elements:
      8
~~Sorted list in A.O~~:
   Enter elements:
      8
      7
      6
      5
      4
      3
      2
      1
   Sorted list in A.O:
      1
      2
      3
      4
      5
      6
      7
      8
```

④ Merge Sort.

```
# include <stdio.h>
# include <conio.h>
# include <stdlib.h>
void merge sort (int *, int);
void merge(int *, int, int *, int);
 void main ( )
 {
   int *arr;
   int i, N;
```

③

```
Printf ("Enter no. of Elements :\n");
scanf ("%d",&N);
arr = (int *) malloc(size(int)*N);

 Printf ("Enter %d elements for Sorting :\n", N);
 for (i=0; i<N ;i++)
 {scanf ("%d", &arr[i]);
 mergesort (arr, N);
 Print f(" Sorted elements:\n");
  for (i=0 ; i< N; i++)
 {Printf (" %d\n", arr[i]);
 }

  void mergesort (int *array, int size)
  {
     int mid;
   if (size ==1)
    return;
    else
    {
     mid = size/2;
      ~~/* function call */~~
      mergesort (array, mid);
      mergesort (array+mid, size-mid);
      merge (array, mid, array+mid, size-mid);
    }
 }
void merge(int *a, int s1, int *b, int s2)
{
    int   i, j, k,  *temp_array;
   temp_arr = (int *)malloc ((s1+s2) * size of (int));
     i = j = k = 0;
     while (i <s1 && j<s2)
    temp_arr [k++]  = (a[i] <b[j])
    while (i <s1)
    temp_arr [k++] = a[i++];
    while (i<s1)
  temp_arr [k++] = a[i++];
  while (j <s2)
  temp_arr [k++] = b[j++];
  for (i=0; i<k; i++)
  a[i] = temp_arr[i];
  }

}
```

| Input | Out Put : |
|---|---|
| Enter no. of Elements: | The Sorted Elements are: |
| 3 | 1 |
| Enter 3 elements to sorting: | 2 |
| 1 | 3 |
| 3 | |
| 2 | |