SORTING

Sorting is the technique in which the elements are stored in a logical order (ascending or descending order). It is basically used for faster searching operations. There are a number of sorting techniques present. Some of the common sorting techniques are:

- bubble sort
- insertion sort
- selection sort
- exchange selection sort
- merge sort
- quick sort
- heap sort
- binary tree sort
- address calculation sort
- radix sort

These sorting techniques returns unique output but differs by their algorithm and complexity (time and space complexity). Time complexity deals with the total time of execution of programming statements where as, space complexity deals with the total memory space needed for storage of algorithm. Some of the techniques are explained below:

1. Program to sort elements using bubble sort.

```
#include <stdio.h>
#define MAX 20
void main( ){
    int arr[MAX], i, j, n, temp;
    printf("Enter the total number of elements of array : ");
    scanf("%d", &n);
    printf("\nEnter the elements of array : ");
    for (i = 0; i < n; i++)
        scanf("%d",&arr[i]);
    printf("\nUnsorted list is :\n");
    for (i = 0; i < n; i++)
        printf("%d\t ", arr[i]);

    /* Bubble Sort Section*/
    for (i = 0; i < n-1 ; i++){
        for (j = 0; j <= n-i-1; j++){
            if (arr[j] > arr[j+1]){
                temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }/*End of if*/
        }/*End of inner for loop*/
    }
    printf("\nAfter sorting the elements the output is : ");
    for (i = 0; i < n; i++)
        printf("%d\t", arr[i]);
}
```

2. Program to sort elements using insertion sort.

```
#include <stdio.h>
#define MAX 20
void main( ){
      int arr[MAX],i,j,k,n;
      printf("Enter the total number of elements of array : ");
      scanf("%d",&n);
      printf("\nEnter the elements of array : ");
      for (i = 0; i < n; i++)
          scanf("%d", &arr[i]);
      printf("\nUnsorted list is :\n");
      for (i = 0; i < n; i++)
           printf("%d\t", arr[i]);
       /*Insertion sort*/
       for( j = 1 ; j<n ; j++ ){
            k=arr[j]; /*k is to be inserted at proper place*/
            for(i=j-1;i>=0 && k<arr[i];i--)
                 arr[ i + 1 ] = arr[ i ];
            arr[ i + 1 ] = k;
       }
       printf("\nThe Sorted list is :  ");
       for (i = 0; i < n; i++)
            printf("%d\t ", arr[i]);
}
```

3. Program to sorting elements using selection sort.

```
#include <stdio.h>
#define MAX 20
void main( ){
      int arr[MAX], i,j,k,n,temp,smallest;
      printf("\nEnter the total number of elements of array  :  ");
      scanf("%d", &n);
      for ( i = 0 ; i < n ;  i++ )
          scanf("%d", &arr[i]);
      printf("Unsorted list is : \n");
      for (i = 0; i < n; i++)
          printf("%d\t", arr[i]);

      /*Selection sort*/
      for(i = 0; i< n - 1 ; i++){  /*Find the smallest element*/
          smallest = i;
          for(k = i + 1; k < n ; k++){
                if(arr[smallest] > arr[k])
                    smallest = k ;
          }
          if( i != smallest ){
                temp = arr [i];
                arr[i] = arr[smallest];
                arr[smallest] = temp ;
          }
```

```
    }/*End of for*/
    printf("\nThe Sorted list is : ");
    for (i = 0; i < n; i++)
        printf("%d \t", arr[i]);
}
```

MERGING

When two sorted arrays are combined to create single sorted array is known as sorting. The first elements of both the sequences are compared and the smaller element is stored in the merge array. The process is applied until one of the array in not complete. Finally, the rest elements are joined at the end of merge array.

Program of merging two sorted arrays into a third sorted array

```
#include<stdio.h>
void main( ){
    int arr1[20], arr2[20], arr3[40];
    int i, j, k;
    int max1, max2;
    printf("Enter the total number of elements of list1 : ");
    scanf("%d", &max1);
    printf("\nTake the elements in sorted order :\n");
    for( i=0 ; i<max1 ; i++ )
        scanf("%d", &arr1[i]);
    printf("\nEnter the total number of elements of list2 : ");
    scanf("%d", &max2);
    printf("\nTake the elements in sorted order :\n");
    for( i=0 ; i<max2 ; i++ )
        scanf("%d",&arr2[i]);
    /* Merging Sort Section*/
    i=0;        /*Index for first array*/
    j=0;        /*Index for second array*/
    k=0;        /*Index for merged array*/
    while( (i < max1) && (j < max2) ){
        if( arr1[i] < arr2[j] )
            arr3[ k++]=arr1[i++];
        else
            arr3[ k++]=arr2[ j++];
    }
    /*Put remaining elements of arr1 into arr3*/
    while( i < max1 )
        arr3[k++]=arr1[i++];
    /*Put remaining elements of arr2 into arr3*/
    while( j < max2 )
        arr3[k++]=arr2[j++];
    printf("\nMerged list output is  :  ");
   for(i=0 ; i<max1+max2 ; i++)
        printf("%d ", arr3[i]);
}
```