

## CODE for Selection Sort: -

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

*Output obtained from the above code for Selection Sort:*

```
C:\Users\Manav Dubey\Desktop>gcc exp1.c
C:\Users\Manav Dubey\Desktop>a
Enter number of elements
5
Enter 5 integers
1492
231
587
10
484
Sorted list in ascending order:
10
231
484
587
1492
C:\Users\Manav Dubey\Desktop>
```

## CODE for Insertion Sort: -

```
#include <stdio.h>
#include <math.h>
void insertionSort(int arr[], int n)
{
    int i, key, j,temp;
    for (i = 1; i < n; i++)
    {
        j = i;
        key = arr[j];
        while (j > 0 && arr[j-1] > key)
        {
            temp=arr[j];
            arr[j]=arr[j-1];
            arr[j-1]=temp;
            j = j-1;
        }
    }
}
void main()
{
    int arr[100],i,n;
    printf("ENTER THE NO. OF ELEMENTS: \n");
    scanf("%d", &n);
    printf("ENTER THE ELEMENTS: \n");
    for (i=0; i < n; i++)
    {
        scanf("%d",&arr[i]);
    }
    insertionSort(arr, n);
    printf("THE SORTED ARRAY IS: \n");
    for (i=0; i < n; i++)
    {
        printf("%d \n", arr[i]);
    }
}
```

*Output obtained from the above code for Insertion Sort:*

```
C:\Users\ Mandavi Dubey\Desktop>gcc exp1.c
C:\Users\ Mandavi Dubey\Desktop>a
ENTER THE NO. OF ELEMENTS:
5
ENTER THE ELEMENTS:
24
233
10
492
9
THE SORTED ARRAY IS:
9
10
24
233
492
C:\Users\ Mandavi Dubey\Desktop>
```

## CODE for Merge Sort: -

```
#include<stdlib.h>
#include<conio.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[100], R[100];

    for (i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];

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

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

void main()
{
    int arr[100],i,a,n;
    printf("ENTER THE NO. OF ELEMENTS: \n");
    scanf("%d",&n);
    printf("ENTER THE ELEMENTS: \n");
    for(i=0;i<n;i++)
        scanf("%d", &arr[i]);
    printf("Given array is \n");
    printArray(arr,n);
    mergeSort(arr, 0, n - 1);
    printf("\nSorted array is \n");
    printArray(arr, n);
}

```

*Output obtained from the above code for Merge Sort:*

```

C:\Users\Mandavi Dubey\Desktop>gcc ex1.c

C:\Users\ Mandavi Dubey \Desktop>a
ENTER THE NO. OF ELEMENTS:
5
ENTER THE ELEMENTS:
24
332
12
3
765
Given array is
24 332 12 3 765

Sorted array is
3 12 24 332 765

```

## CODE for Quick Sort: -

```
#include <conio.h>
#include <stdio.h>
void quick_sort(int[],int,int);
int partition(int[],int,int);
void main()
{
    int a[50],n,i;
    printf("How many elements?\n");
    scanf("%d",&n);
    printf("Enter array elements:\n");
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    quick_sort(a,0,n-1);
    printf("Array after sorting using Quick Sort is:\n");
    for(i=0;i<n;i++)
        printf("%d ",a[i]);
    getch();
}

void quick_sort(int a[],int l,int u)
{
    int j;
    if(l<u)
    {
        j=partition(a,l,u);
        quick_sort(a,l,j-1);
        quick_sort(a,j+1,u);
    }
}

int partition(int a[],int l,int u)
{
    int v,i,j,temp;
    v=a[l];
    i=l;
    j=u+1;
    do
    {
        do
            i++;
        while(a[i]<v&&i<=u);
        do
            j--;
        while(v>a[j]);
        if(i<j){
            temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }
    }while(i<j);
    a[l]=a[i];
    a[i]=v;
    return(i);
}
```

*Output obtained from the above code for Quick Sort:*

```
C:\Users\██████████\Desktop>gcc exp1.c
```

```
C:\Users\██████████\Desktop>a
```

```
How many elements?
```

```
5
```

```
Enter array elements:
```

```
147
```

```
23
```

```
156
```

```
1560
```

```
01
```

```
Array after sorting using Quick Sort is:
```

```
1 23 147 156 1560
```