**PROGRAM**

**Merge Sort:**

#include <stdio.h>

#include <stdlib.h>

void Array(int A[], int size){

int i;

for (i = 0; i < size; i++)

printf("%d ", A[i]);

printf("\n");

}

void merge(int arr[], int l, int m, int h){

int i, j, k;

int n1 = m - l + 1;

int n2 = h - 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 + 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++;

}

printf("Left elements:");

Array(L, n1);

printf("Right elements:");

Array(R, n2);

printf("\n");

}

void mergeSort(int arr[], int l, int h){

if (l < h) {

int m = (l + h) / 2;

mergeSort(arr, l, m);

mergeSort(arr, m + 1, h);

merge(arr, l, m, h);

}

}

int main(){

int n;

printf("How many elements you want: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements: ");

for(int i=0; i<n; i++){

scanf("%d", &arr[i]);

}

mergeSort(arr, 0, n - 1);

printf("\nSorted array is \n");

Array(arr, n);

return 0;

}

**OUTPUT:**

****

**QUICK SORT:**

#include <stdio.h>

void swap(int \*a, int \*b) {

int t = \*a;

\*a = \*b;

\*b = t;

}

int partition(int array[], int low, int high) {

int pivot = array[low];

int i = low;

int j = high;

while(i<j){

do{

i++;

}while(array[i]<=pivot);

do{

j--;

}while(array[j]>pivot);

if(i<j)

swap(&array[i], &array[j]);

}

swap(&array[low], &array[j]);

return (j);

}

void quickSort(int array[], int low, int high){

if (low < high) {

int j=partition(array, low, high);

quickSort(array, low, j);

quickSort(array, j + 1, high);

}

}

void Array(int array[], int size){

for (int i = 0; i < size; ++i) {

printf("%d ", array[i]);

}

printf("\n");

}

int main(){

int n;

printf("How many elements you want: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements: ");

for(int i=0; i<n; i++){

scanf("%d", &arr[i]);

}

printf("Unsorted Array is \n");

Array(arr, n);

quickSort(arr, 0, n);

printf("Sorted array in ascending order is \n");

Array(arr, n);

}

**OUTPUT:**

