1. **Merge sort**

**Code:**

#include <stdio.h>

// Merge function to merge two subarrays

void merge(int arr[], int left, int mid, int right) {

int i, j, k;

int n1 = mid - left + 1;

int n2 = right - mid;

// Create temporary arrays

int L[n1], R[n2];

// Copy data to temporary arrays L[] and R[]

for (i = 0; i < n1; i++)

L[i] = arr[left + i];

for (j = 0; j < n2; j++)

R[j] = arr[mid + 1 + j];

// Merge the temporary arrays back into arr[left..right]

i = 0;

j = 0;

k = left;

while (i < n1 && j < n2) {

if (L[i] <= R[j]) {

arr[k] = L[i];

i++;

} else {

arr[k] = R[j];

j++;

}

k++;

}

// Copy the remaining elements of L[], if any

while (i < n1) {

arr[k] = L[i];

i++;

k++;

}

// Copy the remaining elements of R[], if any

while (j < n2) {

arr[k] = R[j];

j++;

k++;

}

}

// Merge Sort function

void mergeSort(int arr[], int left, int right) {

if (left < right) {

// Find the middle point

int mid = left + (right - left) / 2;

// Sort first and second halves

mergeSort(arr, left, mid);

mergeSort(arr, mid + 1, right);

// Merge the sorted halves

merge(arr, left, mid, right);

}

}

// Function to print an array

void printArray(int arr[], int size) {

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

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

printf("\n");

}

// Main function

int main() {

int arr[100], n;

printf("Enter the number of elements: ");

scanf("%d", &n);

printf("Enter the elements:\n");

for (int 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);

return 0;

}

**Output:**

Enter the number of elements: 5

Enter the elements:

55

11

99

77

33

Given array is

55 11 99 77 33

Sorted array is

11 33 55 77 99

--------------------------------

Process exited after 11.78 seconds with return value 0

Press any key to continue . . .

