**MERGE SORT**

**CODE:**

**#include <stdio.h>**

**#include <stdlib.h>**

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

**int i, j, k;**

**int n1 = m - l + 1;**

**int n2 = r - m;**

**// Create temp arrays**

**int \*L = (int \*)malloc(n1 \* sizeof(int));**

**int \*R = (int \*)malloc(n2 \* sizeof(int));**

**// Copy data to temp arrays L[] and R[]**

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

**L[i] = arr[l + i];**

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

**R[j] = arr[m + 1 + j];**

**// Merge the temp arrays back into arr[l..r]**

**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++;**

**}**

**// 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++;**

**}**

**// Free dynamically allocated memory**

**free(L);**

**free(R);**

**}**

**void mergeSort(int arr[], int l, int r) {**

**if (l < r) {**

**int m = l + (r - l) / 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");**

**}**

**int main() {**

**int arr[] = { 12, 11, 13, 5, 6, 7 };**

**int arr\_size = sizeof(arr) / sizeof(arr[0]);**

**printf("Given array is \n");**

**printArray(arr, arr\_size);**

**mergeSort(arr, 0, arr\_size - 1);**

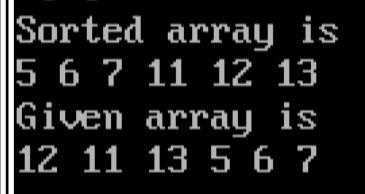
**printf("\nSorted array is \n");**

**printArray(arr, arr\_size);**

**return 0;**

**}**

**OUTPUT:**

****