Merge sort

**#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;**

**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++;**

**}**

**}**

**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);**

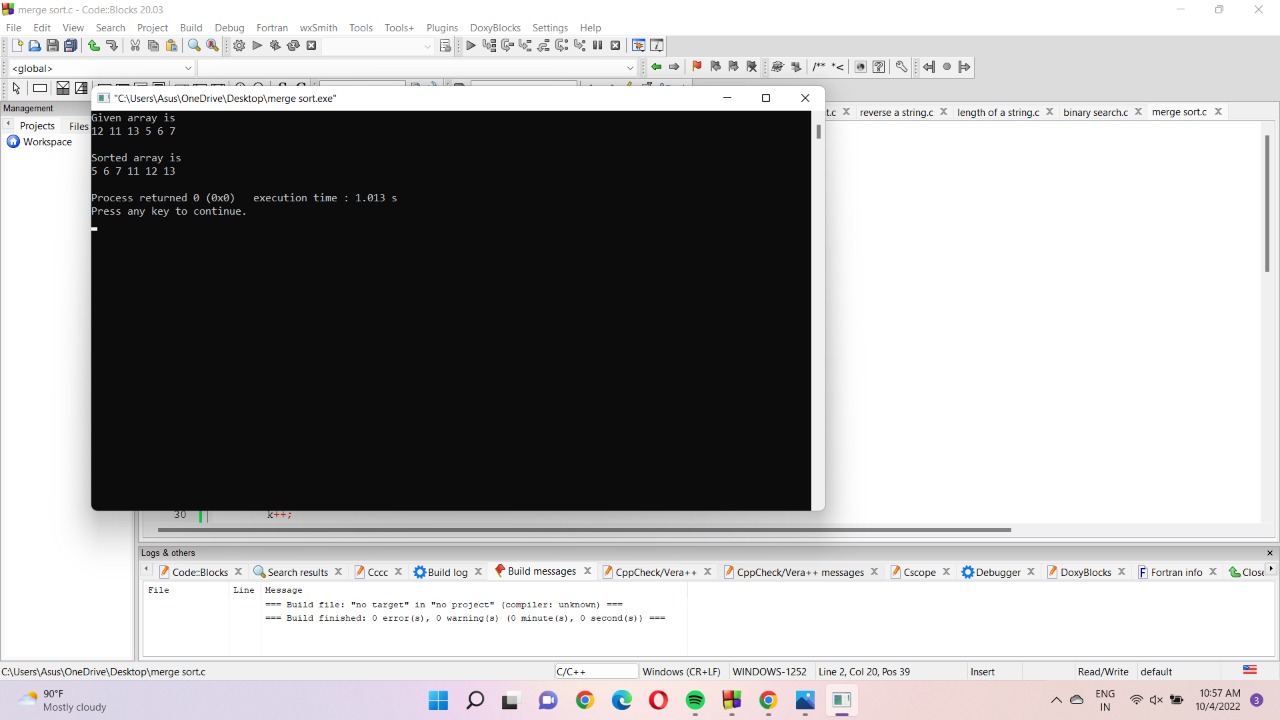
**mergeSort(arr, 0, arr\_size – 1);**

**printf(“\nSorted array is \n”);**

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

**return 0;**

**}**

****