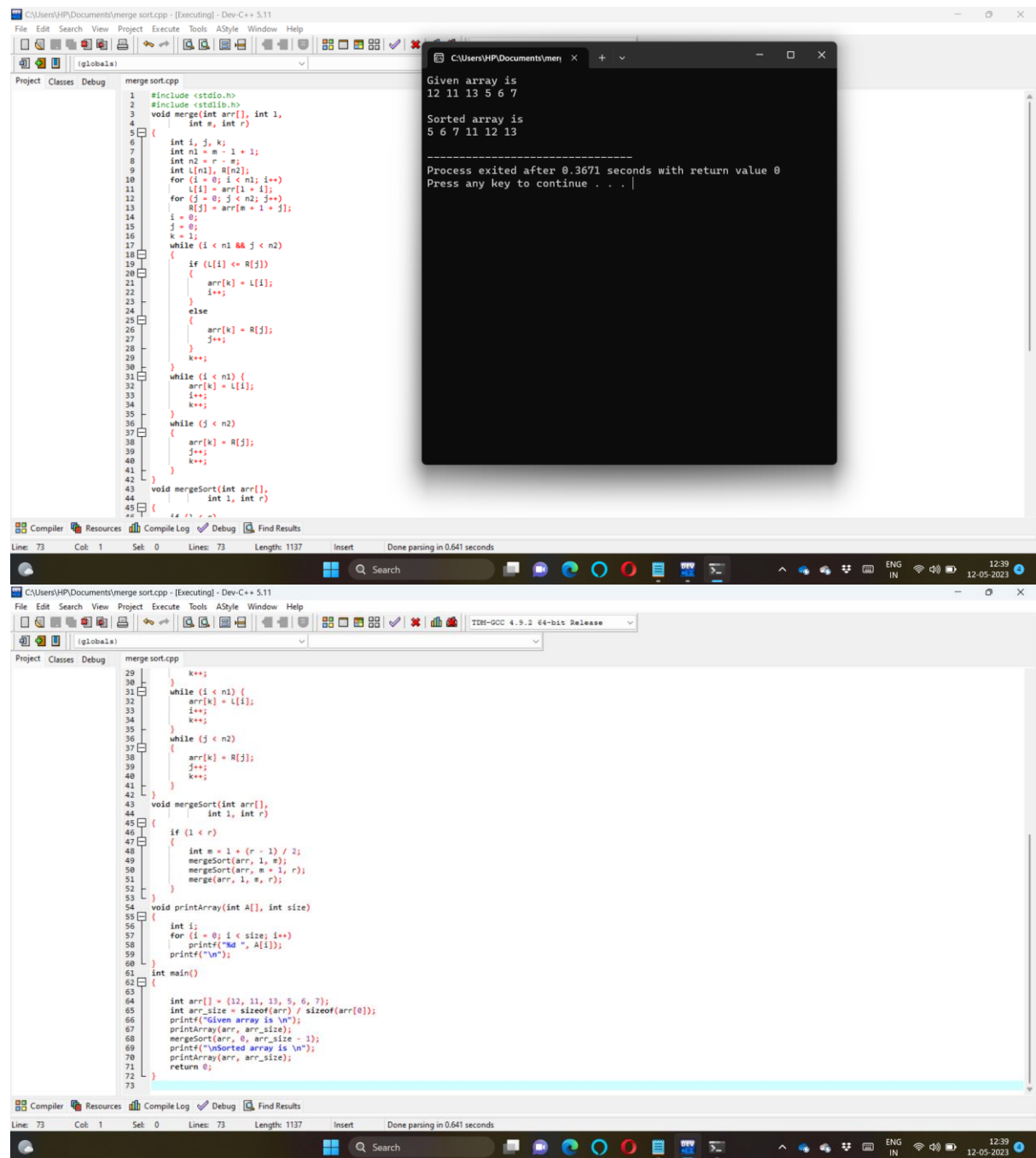


NAME:P.HARI HASSAN

REG.NO:192210633

DATE:12/05/23

1.C PROGRAMM FOR MERGE SORT



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 void merge(int arr[], int l,
4             int m, int r)
5 {
6     int i, j, k;
7     int n1 = m - l + 1;
8     int n2 = r - m;
9     int L[n1], R[n2];
10    for (i = 0; i < n1; i++)
11        L[i] = arr[l + i];
12    for (j = 0; j < n2; j++)
13        R[j] = arr[m + 1 + j];
14    i = 0;
15    j = 0;
16    k = l;
17    while (i < n1 && j < n2)
18    {
19        if (L[i] <= R[j])
20        {
21            arr[k] = L[i];
22            i++;
23        }
24        else
25        {
26            arr[k] = R[j];
27            j++;
28        }
29        k++;
30    }
31    while (i < n1)
32    {
33        arr[k] = L[i];
34        i++;
35        k++;
36    }
37    while (j < n2)
38    {
39        arr[k] = R[j];
40        j++;
41        k++;
42    }
43 }
44 void mergeSort(int arr[],
45                 int l, int r)
46 {
47     if (l < r)
48     {
49         int m = l + (r - l) / 2;
50         mergeSort(arr, l, m);
51         mergeSort(arr, m + 1, r);
52         merge(arr, l, m, r);
53     }
54 }
55 void printArray(int A[], int size)
56 {
57     int i;
58     for (i = 0; i < size; i++)
59         printf("%d ", A[i]);
60     printf("\n");
61 }
62 int main()
63 {
64     int arr[] = {12, 11, 13, 5, 6, 7};
65     int arr_size = sizeof(arr) / sizeof(arr[0]);
66     printf("Given array is\n");
67     printArray(arr, arr_size);
68     mergeSort(arr, 0, arr_size - 1);
69     printf("\nSorted array is\n");
70     printArray(arr, arr_size);
71     return 0;
72 }
73 }
```

```
Given array is
12 11 13 5 6 7

Sorted array is
5 6 7 11 12 13

Process exited after 0.3671 seconds with return value 0
Press any key to continue . . .
```