

# Rajalakshmi Engineering College

Name: bhagawath narayanan n  
Email: 241501034@rajalakshmi.edu.in  
Roll no: 241501034  
Phone: 6374835866  
Branch: REC  
Department: I AIML AD  
Batch: 2028  
Degree: B.E - AI & ML

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

### REC\_DS using C\_Week 6\_COD\_Question 5

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

Jose has an array of N fractional values, represented as double-point numbers. He needs to sort these fractions in increasing order and seeks your help.

Write a program to help Jose sort the array using the merge sort algorithm.

##### ***Input Format***

The first line of input consists of an integer N, representing the number of fractions to be sorted.

The second line consists of N double-point numbers, separated by spaces, representing the fractions array.

##### ***Output Format***

The output prints N double-point numbers, sorted in increasing order, and rounded to three decimal places.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 4

0.123 0.543 0.321 0.789

Output: 0.123 0.321 0.543 0.789

### **Answer**

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int compare(double a, double b) {  
    return a < b;  
}
```

```
// Merge function for merge sort
```

```
void merge(double arr[], int l, int m, int r) {  
    int n1 = m - l + 1;  
    int n2 = r - m;
```

```
    double L[10], R[10]; // Max N = 10 as per constraints
```

```
    for (int i = 0; i < n1; i++)  
        L[i] = arr[l + i];
```

```
    for (int j = 0; j < n2; j++)  
        R[j] = arr[m + 1 + j];
```

```
    int i = 0, j = 0, k = l;
```

```
    // Merge back into original array in sorted order
```

```
    while (i < n1 && j < n2) {  
        if (compare(L[i], R[j])) {  
            arr[k++] = L[i++];  
        } else {  
            arr[k++] = R[j++];
```

```

    }
}

// Copy remaining elements
while (i < n1)
    arr[k++] = L[i++];

while (j < n2)
    arr[k++] = R[j++];
}

// Merge Sort function
void mergeSort(double arr[], int l, int r) {
    if (l < r) {
        int m = l + (r - l) / 2;

        mergeSort(arr, l, m); // Sort left half
        mergeSort(arr, m + 1, r); // Sort right half
        merge(arr, l, m, r); // Merge the sorted halves
    }
}

int main() {
    int n;
    scanf("%d", &n);
    double fractions[n];
    for (int i = 0; i < n; i++) {
        scanf("%lf", &fractions[i]);
    }
    mergeSort(fractions, 0, n - 1);
    for (int i = 0; i < n; i++) {
        printf("%.3f ", fractions[i]);
    }
    return 0;
}

```

**Status :** Correct

**Marks :** 10/10