

# Rajalakshmi Engineering College

Name: naveen prasath  
Email: 240701352@rajalakshmi.edu.in  
Roll no: 240701352  
Phone: 9585322006  
Branch: REC  
Department: I CSE FD  
Batch: 2028  
Degree: B.E - CSE

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>
```

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

```
    int n1 = mid - left + 1;
```

```
    int n2 = right - mid;
```

```
    double L[n1], R[n2];
```

```
    for (int i = 0; i < n1; i++) {
```

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

```
    }
```

```
    for (int i = 0; i < n2; i++) {
```

```
        R[i] = arr[mid + 1 + i];
```

```
    }
```

```
    int 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++;
```

```
    }
```

```

while (i < n1) {
    arr[k] = L[i];
    i++;
    k++;
}

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

void mergeSort(double arr[], int left, int right) {
    if (left < right) {
        int mid = left + (right - left) / 2;
        mergeSort(arr, left, mid);
        mergeSort(arr, mid + 1, right);
        merge(arr, left, mid, right);
    }
}

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