

Rajalakshmi Engineering College

Name: akas prabhu
Email: 240801017@rajalakshmi.edu.in
Roll no: 240801017
Phone: 9360484615
Branch: REC
Department: I ECE FA
Batch: 2028
Degree: B.E - ECE

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>
```

```
// Comparison function
```

```
int compare(double a, double b) {
```

```
    if (a < b) return -1;
```

```
    if (a > b) return 1;
```

```
    return 0;
```

```
}
```

```
// Merge function
```

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

```
    int left_size = m - l + 1;
```

```
    int right_size = r - m;
```

```
    double left[left_size], right[right_size];
```

```
    // Copy data to temporary arrays
```

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

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

```
    }
```

```
    for (int j = 0; j < right_size; j++) {
```

```
        right[j] = arr[m + 1 + j];
```

```
    }
```

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

```

// Merge the two sorted subarrays
while (i < left_size && j < right_size) {
    if (compare(left[i], right[j]) <= 0) {
        arr[k++] = left[i++];
    } else {
        arr[k++] = right[j++];
    }
}

// Copy the remaining elements of left[] if any
while (i < left_size) {
    arr[k++] = left[i++];
}

// Copy the remaining elements of right[] if any
while (j < right_size) {
    arr[k++] = right[j++];
}
}

```

```

// Merge sort function
void mergeSort(double arr[], int l, int r) {
    if (l < r) {
        int m = l + (r - l) / 2; // Correct midpoint
        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);
        merge(arr, l, m, r);
    }
}

```

```

// Utility function to print array
void printArray(double arr[], int n) {
    for (int i = 0; i < n; i++) {
        printf("%.3f ", arr[i]);
    }
    printf("\n");
}

```

```

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