Rajalakshmi Engineering College

Name: suriya SM
Email: 241801284@rajalakshmi.edu.in
Roll no: 241801284
Phone: 8110855156
Branch: REC
Department: I AI & DS FD
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 6_COD_Question 1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

John and Mary are collaborating on a project that involves data analysis. They each have a set of age data, one sorted in ascending order and the other in descending order. However, their analysis requires the data to be in ascending order.

Write a program to help them merge the two sets of age data into a single sorted array in ascending order using merge sort.

Input Format

The first line of input consists of an integer N, representing the number of age values in each dataset.

The second line consists of N space-separated integers, representing the ages of participants in John's dataset (in ascending order).

The third line consists of N space-separated integers, representing the ages of participants in Mary's dataset (in descending order).

Output Format

The output prints a single line containing space-separated integers, which represents the merged dataset of ages sorted in ascending order.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
1 3 5 7 9
10 8 6 4 2
Output: 1 2 3 4 5 6 7 8 9 10

Answer

```
#include <stdio.h>

// You are using GCC
#include <stdio.h>

void merge_sorted_arrays(int arr1[], int arr2[], int merged[], int N) {
    int i = 0, j = N - 1, k = 0;


    while (i < N && j >= 0) {
        if (arr1[i] < arr2[j]) {
            merged[k++] = arr1[i++];
        } else {
            merged[k++] = arr2[j--];
        }
    }


    while (i < N) {
        merged[k++] = arr1[i++];
    }
```

```
    while (j >= 0) {
        merged[k++] = arr2[j--];
    }
}

int main() {
    int N;
    scanf("%d", &N);

    int arr1[N], arr2[N], merged[2 * N];


    for (int i = 0; i < N; i++) {
        scanf("%d", &arr1[i]);
    }


    for (int i = 0; i < N; i++) {
        scanf("%d", &arr2[i]);
    }


    merge_sorted_arrays(arr1, arr2, merged, N);


    for (int i = 0; i < 2 * N; i++) {
        printf("%d ", merged[i]);
    }

    return 0;
}
int main() {
    int n, m;
    scanf("%d", &n);
    int arr1[n], arr2[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr1[i]);
    }
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr2[i]);
    }
```

```
    int merged[n + n];
    mergeSort(arr1, n);
    mergeSort(arr2, n);
    merge(merged, arr1, arr2, n, n);
    for (int i = 0; i < n + n; i++) {
        printf("%d ", merged[i]);
    }
    return 0;
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: suriya SM
Email: 241801284@rajalakshmi.edu.in
Roll no: 241801284
Phone: 8110855156
Branch: REC
Department: I AI & DS FD
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 6_COD_Question 3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are the lead developer of a text-processing application that assists writers in organizing their thoughts. One crucial feature is a character-sorting service that helps users highlight the most critical elements of their text.

To achieve this, you decide to enhance the service to sort characters in descending order using the Quick-Sort algorithm. Implement the algorithm to efficiently rearrange the characters, ensuring that it is sorted in descending order.

Input Format

The first line of the input consists of a positive integer value N, representing the number of characters to be sorted.

The second line of input consists of N space-separated lowercase alphabetical characters.

Output Format

The output displays the set of alphabetical characters, sorted in descending order.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5
a d g j k
Output: k j g d a

Answer

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

// You are using GCC
#include <stdio.h>

void swap(char *a, char *b) {
    char temp = *a;
    *a = *b;
    *b = temp;
}

int partition(char arr[], int low, int high) {
    char pivot = arr[high];
    int i = low - 1;

    for (int j = low; j < high; j++) {
        if (arr[j] > pivot) {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[i + 1], &arr[high]);
```

```
    return i + 1;
}

void quick_sort(char arr[], int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);
        quick_sort(arr, low, pi - 1);
        quick_sort(arr, pi + 1, high);
    }
}

int main() {
    int N;
    scanf("%d", &N);

    char arr[N];
    for (int i = 0; i < N; i++) {
        scanf(" %c", &arr[i]);
    }

    quick_sort(arr, 0, N - 1);

    for (int i = 0; i < N; i++) {
        printf("%c ", arr[i]);
    }

    return 0;
}
int main() {
    int n;
    scanf("%d", &n);

    char characters[n];

    for (int i = 0; i < n; i++) {
        char input;
        scanf(" %c", &input);
        characters[i] = input;
    }

    quicksort(characters, 0, n - 1);
```

```
    for (int i = 0; i < n; i++) {
        printf("%c ", characters[i]);
    }

    return 0;
}
```

Status : Correct **Marks : 10/10**

Rajalakshmi Engineering College

Name: suriya SM
Email: 241801284@rajalakshmi.edu.in
Roll no: 241801284
Phone: 8110855156
Branch: REC
Department: I AI & DS FD
Batch: 2028
Degree: B.E - AI & DS

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>

// You are using GCC
#include <stdio.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 j = 0; j < n2; j++)
        R[j] = arr[mid + 1 + j];

    int i = 0, j = 0, k = left;

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

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

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

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

        merge_sort(arr, left, mid);
        merge_sort(arr, mid + 1, right);

        merge(arr, left, mid, right);
    }
}

int main() {
    int N;
    scanf("%d", &N);

    double arr[N];

    for (int i = 0; i < N; i++)
        scanf("%lf", &arr[i]);

    merge_sort(arr, 0, N - 1);

    for (int i = 0; i < N; i++)
        printf("%.3f ", arr[i]);

    return 0;
}
```

```
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**

Rajalakshmi Engineering College

Name: suriya SM
Email: 241801284@rajalakshmi.edu.in
Roll no: 241801284
Phone: 8110855156
Branch: REC
Department: I AI & DS FD
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 7_COD_Question 3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In a messaging application, users maintain a contact list with names and corresponding phone numbers. Develop a program to manage this contact list using a dictionary implemented with hashing.

The program allows users to add contacts, delete contacts, and check if a specific contact exists. Additionally, it provides an option to print the contact list in the order of insertion.

Input Format

The first line consists of an integer n, representing the number of contact pairs to be inserted.

Each of the next n lines consists of two strings separated by a space: the name of the contact (key) and the corresponding phone number (value).

The last line contains a string k, representing the contact to be checked or removed.

Output Format

If the given contact exists in the dictionary:

1. The first line prints "The given key is removed!" after removing it.
2. The next n - 1 lines print the updated contact list in the format: "Key: X; Value: Y" where X represents the contact's name and Y represents the phone number.

If the given contact does not exist in the dictionary:

1. The first line prints "The given key is not found!".
2. The next n lines print the original contact list in the format: "Key: X; Value: Y" where X represents the contact's name and Y represents the phone number.

Refer to the sample outputs for the formatting specifications.

Sample Test Case

Input: 3
Alice 1234567890
Bob 9876543210
Charlie 4567890123
Bob

Output: The given key is removed!
Key: Alice; Value: 1234567890
Key: Charlie; Value: 4567890123

Answer

```
// You are using
#include <stdio.h>
#include <string.h>

#define MAX_CONTACTS 50
```

```
typedef struct {
    char name[11];
    char phone[11];
} Contact;

int main() {
    int n, i, index = -1;
    char key[11];
    Contact contacts[MAX_CONTACTS];

    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        scanf("%s %s", contacts[i].name, contacts[i].phone);
    }
    scanf("%s", key);

    for (i = 0; i < n; i++) {
        if (strcmp(contacts[i].name, key) == 0) {
            index = i;
            break;
        }
    }

    if (index != -1) {
        printf("The given key is removed!\n");
        for (i = index; i < n - 1; i++) {
            contacts[i] = contacts[i + 1];
        }
        n--;
    } else {
        printf("The given key is not found!\n");
    }

    for (i = 0; i < n; i++) {
        printf("Key: %s; Value: %s\n", contacts[i].name, contacts[i].phone);
    }

    return 0;
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: suriya SM
Email: 241801284@rajalakshmi.edu.in
Roll no: 241801284
Phone: 8110855156
Branch: REC
Department: I AI & DS FD
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 7_COD_Question 4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Develop a program using hashing to manage a fruit contest where each fruit is assigned a unique name and a corresponding score. The program should allow the organizer to input the number of fruits and their names with scores.

Then, it should enable them to check if a specific fruit, identified by its name, is part of the contest. If the fruit is registered, the program should display its score; otherwise, it should indicate that it is not included in the contest.

Input Format

The first line consists of an integer N, representing the number of fruits in the contest.

The following N lines contain a string K and an integer V, separated by a space, representing the name and score of each fruit in the contest.

The last line consists of a string T, representing the name of the fruit to search for.

Output Format

If T exists in the dictionary, print "Key "T" exists in the dictionary.".

If T does not exist in the dictionary, print "Key "T" does not exist in the dictionary.".

Refer to the sample outputs for the formatting specifications.

Sample Test Case

Input: 2
banana 2
apple 1
Banana

Output: Key "Banana" does not exist in the dictionary.

Answer

```
// You are using GCC
#include <stdio.h>
#include <string.h>

#define MAX_FRUITS 15

typedef struct {
    char name[20];
    int score;
} Fruit;

int main() {
    int n, i;
    char key[20];
    Fruit fruits[MAX_FRUITS];
```

```
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        scanf("%s %d", fruits[i].name, &fruits[i].score);
    }
    scanf("%s", key);

    for (i = 0; i < n; i++) {
        if (strcmp(fruits[i].name, key) == 0) {
            printf("Key \"%s\" exists in the dictionary.\n", key);
            return 0;
        }
    }

    printf("Key \"%s\" does not exist in the dictionary.\n", key);
    return 0;
}
```

Status : Correct **Marks : 10/10**