Index-225036C

Wednesday, May 28, 2025   11:18 AM

Write a C program to do the following tasks .

● given two unsorted arrays from the user

● Sorts each array using Selection Sort

● Merges them into a single final sorted array

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

// Function to perform selection sort on an array
void selectionSort(int arr[], int n) {
    int i, j, minIdx, temp;

    for (i = 0; i < n - 1; i++) {
        minIdx = i;

        // Find the minimum element in remaining unsorted array
        for (j = i + 1; j < n; j++) {
            if (arr[j] < arr[minIdx]) {
                minIdx = j;
            }
        }

        // Swap the found minimum element with the first element
        if (minIdx != i) {
            temp = arr[i];
            arr[i] = arr[minIdx];
            arr[minIdx] = temp;
        }
    }
}

// Function to merge two sorted arrays into a single sorted array
void mergeArrays(int arr1[], int n1, int arr2[], int n2, int merged[]) {
    int i = 0, j = 0, k = 0;

    // Merge elements from both arrays in sorted order
    while (i < n1 && j < n2) {
        if (arr1[i] <= arr2[j]) {
            merged[k++] = arr1[i++];
        } else {
            merged[k++] = arr2[j++];
        }
    }

    // Copy remaining elements of arr1, if any
    while (i < n1) {
        merged[k++] = arr1[i++];
    }

    // Copy remaining elements of arr2, if any
    while (j < n2) {
        merged[k++] = arr2[j++];
    }
}

// Function to print an array
void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main() {
    int n1, n2;

    // Get size of first array
    printf("Enter the size of first array: ");
    scanf("%d", &n1);

    int *arr1 = (int*)malloc(n1 * sizeof(int));

    // Input elements for first array
    printf("Enter %d elements for first array: ", n1);
    for (int i = 0; i < n1; i++) {
        scanf("%d", &arr1[i]);
    }

    // Get size of second array
    printf("Enter the size of second array: ");
    scanf("%d", &n2);

    int *arr2 = (int*)malloc(n2 * sizeof(int));

    // Input elements for second array
    printf("Enter %d elements for second array: ", n2);
    for (int i = 0; i < n2; i++) {
        scanf("%d", &arr2[i]);
    }
    // Display original arrays
    printf("\nOriginal first array: ");
    printArray(arr1, n1);
    printf("Original second array: ");
    printArray(arr2, n2);

    // Sort both arrays using selection sort
    selectionSort(arr1, n1);
    selectionSort(arr2, n2);

    // Display sorted arrays
    printf("\nAfter sorting using Selection Sort:\n");
    printf("Sorted first array: ");
    printArray(arr1, n1);
    printf("Sorted second array: ");
    printArray(arr2, n2);

    // Create merged array
    int *merged = (int*)malloc((n1 + n2) * sizeof(int));

    // Merge the sorted arrays
    mergeArrays(arr1, n1, arr2, n2, merged);

    // Display final merged sorted array
    printf("\nFinal merged sorted array: ");
    printArray(merged, n1 + n2);

    // Free dynamically allocated memory
    free(arr1);
    free(arr2);
    free(merged);

    return 0;
}
```