Application Based on Merge Sort

Title : Online Shopping Price Manager

Program Code :

```
#include <stdio.h>

#include <string.h>

#define SIZE 50

struct Product {

        char name[50];

        float price;

};

void merge(struct Product arr[], int left, int mid, int right) {

        int n1 = mid - left + 1;

        int n2 = right - mid;

        struct Product L[n1], R[n2];    // ? Variable Length Arrays
(modern C)

            int i, j, k;

// Copy left half

        for (i = 0; i < n1; i++)

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

// Copy right half
```

```
        for (j = 0; j < n2; j++)
                R[j] = arr[mid + 1 + j];
          i = 0;
        j = 0;
        k = left;
// Merge two halves
        while (i < n1 && j < n2) {
                if (L[i].price <= R[j].price)      // ascending order
                        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++];
}
void mergeSort(struct Product arr[], int left, int right) {
        if (left < right) {
```

```
        int mid = (left + right) / 2;
        mergeSort(arr, left, mid);
        mergeSort(arr, mid + 1, right);
        merge(arr, left, mid, right);
    }
}
int main()
{
    struct Product arr[SIZE];
    int n, i;
    printf("Enter number of products (max %d): ", SIZE);
    scanf("%d", &n);
  for (i = 0; i < n; i++) {
        printf("\nEnter details for Product %d\n", i + 1);
        printf("Name: ");
        scanf(" %[^\n]", arr[i].name);
        printf("Price: ");
        scanf("%f", &arr[i].price);
    }
mergeSort(arr, 0, n - 1);
```

```
    printf("\nProducts sorted by Price (lowest first):\n");
        printf("Name\t\tPrice\n");
        for (i = 0; i < n; i++) {
            printf("%s\t\t%.2f\n", arr[i].name, arr[i].price);
        }
return 0;
}
```

Output :