

DAA 3

```
#include <iostream>

using namespace std;

// Function to merge two sorted subarrays
void merge(int arr[], int left, int mid, int right) {
    int n1 = mid - left + 1;
    int n2 = right - mid;

    // Create temporary arrays
    int leftArr[n1], rightArr[n2];

    // Copy data to temporary arrays
    for (int i = 0; i < n1; i++) {
        leftArr[i] = arr[left + i];
    }
    for (int j = 0; j < n2; j++) {
        rightArr[j] = arr[mid + 1 + j];
    }

    // Merge the temporary arrays
    int i = 0, j = 0, k = left;
    while (i < n1 && j < n2) {
        if (leftArr[i] <= rightArr[j]) {
            arr[k] = leftArr[i];
            i++;
        }
```

```
    } else {  
        arr[k] = rightArr[j];  
        j++;  
    }  
    k++;  
}
```

```
// Copy remaining elements, if any
```

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

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

```
// Function to implement merge sort using divide and conquer approach
```

```
void mergeSort(int arr[], int left, int right) {
```

```
    if (left < right) {  
        int mid = left + (right - left) / 2;
```

```
        // Recursively sort the left and right subarrays
```

```
        mergeSort(arr, left, mid);
```

```

mergeSort(arr, mid + 1, right);

// Merge the sorted subarrays
merge(arr, left, mid, right);
}
}

// Function to print the array
void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}

// Function to get input from user
void getInput(int arr[], int size) {
    cout << "Enter " << size << " elements: ";
    for (int i = 0; i < size; i++) {
        cin >> arr[i];
    }
}

// Example usage
int main() {
    int n;
    cout << "Enter the number of elements: ";

```

```
cin >> n;
```

```
int* arr = new int[n];
```

```
getInput(arr, n);
```

```
cout << "Original array: ";
```

```
printArray(arr, n);
```

```
mergeSort(arr, 0, n - 1);
```

```
cout << "Sorted array: ";
```

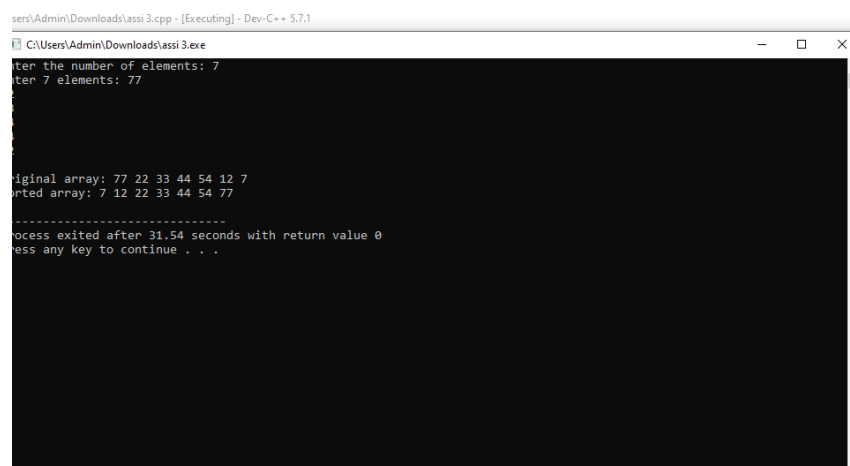
```
printArray(arr, n);
```

```
delete[] arr;
```

```
return 0;
```

```
}
```

Output:



```
sers\Admin\Downloads\assi 3.cpp - [Executing] - Dev-C++ 5.7.1
C:\Users\Admin\Downloads\assi 3.exe
Enter the number of elements: 7
Enter 7 elements: 77

Original array: 77 22 33 44 54 12 7
Sorted array: 7 12 22 33 44 54 77

-----
Process exited after 31.54 seconds with return value 0
Press any key to continue . . .
```