	Marwadi University
	Marwadi University **Faculty of Engineering and Technology** **Department of Information and Communication Technology**

Subject: Design and Analysis of Algorithm	**Aim:** Implementing Sorting Algorithm using Divide and Conquer Approach	
Experiment No: 04	**Date: 13/09/2025**	**Enrollment No: 92301733049**

AIM: **Implementing Sorting Algorithm using Divide and Conquer Approach**

I.   Merge Sort

```cpp
#include <iostream>
using namespace std;

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

    // Temporary arrays
    int L[n1], R[n2];

    // Copy data to temp arrays
    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];

    // Merge the two halves
    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++];
    }

    // Copy remaining elements
    while (i < n1) arr[k++] = L[i++];
    while (j < n2) arr[k++] = R[j++];
}

// Recursive merge sort
void mergeSort(int arr[], int left, int right) {
    if (left < right) {
        int mid = (left + right) / 2;
        mergeSort(arr, left, mid);
```

	Marwadi University **Faculty of Engineering and Technology** **Department of Information and Communication Technology**
Subject: Design and Analysis of Algorithm	**Aim:** Implementing Sorting Algorithm using Divide and Conquer Approach
Experiment No: 04	**Date: 13/09/2025**	**Enrollment No: 92301733049**

```cpp
        mergeSort(arr, mid + 1, right);
        merge(arr, left, mid, right);
    }
}

int main() {
    int arr[] = {38, 27, 43, 3, 9, 82, 10};
    int n = sizeof(arr) / sizeof(arr[0]);

    cout << "Original array: ";
    for (int i = 0; i < n; i++) cout << arr[i] << " ";
    cout << endl;

    mergeSort(arr, 0, n - 1);

    cout << "Sorted array (Merge Sort): ";
    for (int i = 0; i < n; i++) cout << arr[i] << " ";
    cout << endl;

    return 0;
}
```
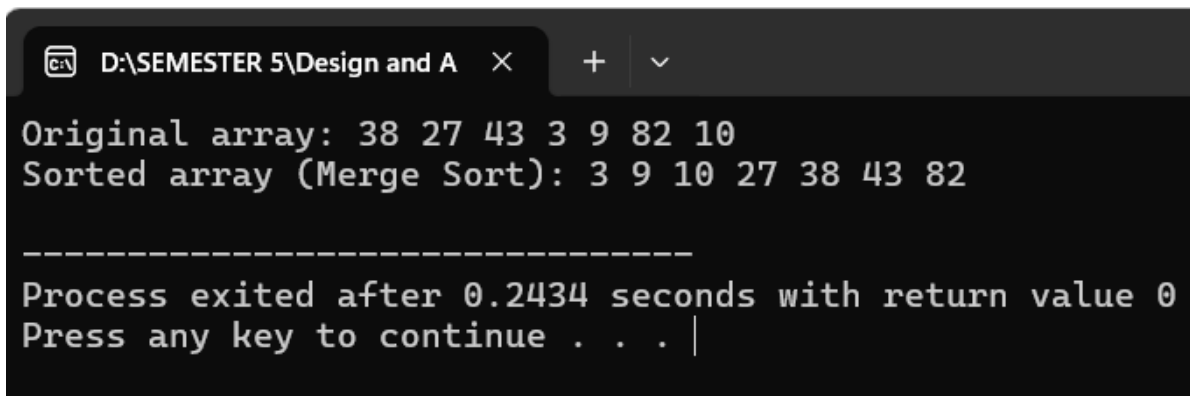
	Marwadi University **Faculty of Engineering and Technology** **Department of Information and Communication Technology**
Subject: Design and Analysis of Algorithm	**Aim: Implementing Sorting Algorithm using Divide and Conquer Approach**
Experiment No: 04	**Date: 13/09/2025**	**Enrollment No: 92301733049**

II.   Quick Sort

```cpp
#include <iostream>
using namespace std;

// Partition function
int partition(int arr[], int low, int high) {
    int pivot = arr[high];  // last element as pivot
    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);
}

// Recursive QuickSort
void quickSort(int arr[], int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

int main() {
    int arr[] = {10, 7, 8, 9, 1, 5};
    int n = sizeof(arr) / sizeof(arr[0]);

    cout << "Original array: ";
    for (int i = 0; i < n; i++) cout << arr[i] << " ";
    cout << endl;

    quickSort(arr, 0, n - 1);

    cout << "Sorted array (Quick Sort): ";
    for (int i = 0; i < n; i++) cout << arr[i] << " ";
```
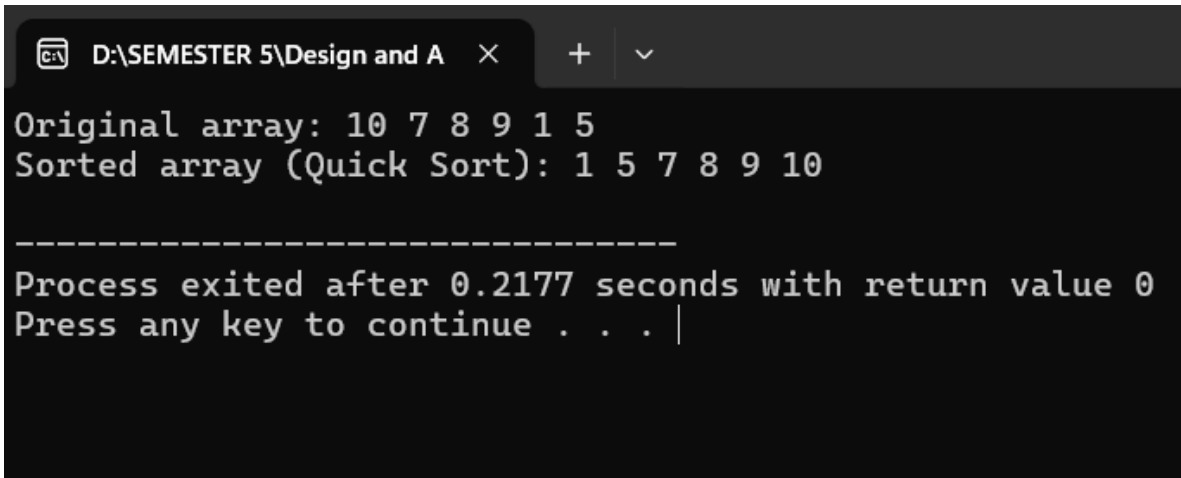
![Marwadi University logo]	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**
Subject: Design and Analysis of Algorithm	**Aim: Implementing Sorting Algorithm using Divide and Conquer Approach**
Experiment No: 04	**Date: 13/09/2025**	**Enrollment No: 92301733049**

```
    cout << endl;

    return 0;
}
```