**Task No. 1:**

Implement merge sort algorithm to merge two integer arrays into a third array in sorted order.

**Solution:**

class Program

{

static void sortedMerge(int[] a, int[] b,

int[] res, int n, int m)

{

// Sorting a[] and b[]

Array.Sort(a);

Array.Sort(b);

// Merge two sorted arrays into res[]

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

while (i < n && j < m)

{

if (a[i] <= b[j])

{

res[k] = a[i];

i += 1;

k += 1;

}

else

{

res[k] = b[j];

j += 1;

k += 1;

}

}

while (i < n)

{

// Merging remaining

// elements of a[] (if any)

res[k] = a[i];

i += 1;

k += 1;

}

while (j < m)

{

// Merging remaining

// elements of b[] (if any)

res[k] = b[j];

j += 1;

k += 1;

}

}

/\* Driver program to test

above function \*/

public static void Main()

{

int[] a = { 7, 3, 19 };

int[] b = { 5, 1, 25, 11 };

int n = a.Length;

int m = b.Length;

// Final merge list

int[] res = new int[n + m];

sortedMerge(a, b, res, n, m);

Console.Write("Sorted merged list :");

for (int i = 0; i < n + m; i++)

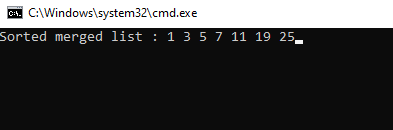
Console.Write(" " + res[i]);

Console.ReadLine();

}

}

**Output:**



**Task No. 2:**

Implement recursive method of merge sort algorithm to sort an array of 10 characters.

**Solution:**

class Program

{

static void CharacterMergeSort(char[] A)

{

int n = A.Length;

if (n < 2)

return;

int mid = n / 2;

char[] LA = new char[mid];

char[] RA = new char[n - mid];

for (int i = 0; i < LA.Length; i++)

LA[i] = A[i];

for (int i = mid; i < n; i++)

RA[i - mid] = A[i];

CharacterMergeSort(LA);

CharacterMergeSort(RA);

CharacterMerge(LA, RA, A);

}

static void CharacterMerge(char[] LA, char[] RA, char[] A)

{

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

while (i < LA.Length && j < RA.Length)

{

int value = LA[i].CompareTo(RA[j]);

if (value < 0)

{

A[k] = LA[i];

i++;

k++;

}

else

{

A[k] = RA[j];

j++;

k++;

}

}

while (i < LA.Length)

{

A[k] = LA[i];

i++;

k++;

}

while (j < RA.Length)

{

A[k] = RA[j];

j++;

k++;

}

}

static void DisplayCharArray(char[] array)

{

for (int i = 0; i < array.Length; i++)

Console.Write(array[i] + " ");

}

static void InputCharArray(char[] A)

{

for (int i = 0; i < A.Length; i++)

{

Console.Write("Enter Character # {0}: ", i + 1);

char.TryParse(Console.ReadLine(), out A[i]);

}

}

public static void Main()

{

int q;

Console.Write("Type Length of character array in integer: ");

int.TryParse(Console.ReadLine(), out q);

char[] p = new char[q];

InputCharArray(p);

CharacterMergeSort(p);

Console.WriteLine("Character Array in sorted order:");

DisplayCharArray(p);

}

}

}

**Output:**

