**Task1: Take numeric input from user and implement recursive method of merge sort algorithm.**

**Code:**

static void Main(string[] args)

{

int[] A ;

Console.WriteLine("Enter length of array");

int N = Convert.ToInt32(Console.ReadLine());

A = new int[N];

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

{

Console.WriteLine("Enter value");

A[i] = Convert.ToInt32(Console.ReadLine());

}

Console.WriteLine();

MergeSort(A, 0, A.Length - 1);

Console.WriteLine("Sorted array");

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

{

Console.WriteLine(A[i]);

}

}

public static void MergeSort(int[] A,int LB, int UB)

{

if (LB < UB)

{

int mid = (LB + UB)/2;

MergeSort(A, LB, mid);

MergeSort(A, mid + 1, UB);

Merge(A, LB, mid, mid + 1, UB);

return;

}

else

{

return;

}

}

public static void Merge(int[] A,int LB1,int UB1,int LB2,int UB2)

{

try

{

int[] C = new int[A.Length];

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

{

C[l] = A[l];

}

int i, j, k;

for (i = LB1, j = LB2, k = LB1; i <=UB1 && j <=UB2 && k <=UB2; k++)

{

if (C[i] <=C[j])

{

A[k] = C[i];

i++;

}

else

{

A[k] = C[j];

j++;

}

}

for (; i <=UB1; i++, k++)

{

A[k] = C[i];

}

for (; j <=UB2; j++, k++)

{

A[k] = C[j];

}

return;

}

catch (Exception)

{

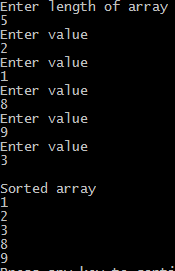
Console.WriteLine("An error occured");

return;

}

}

**Output:**



**Task2: Take string input from user and implement iterative method of merge sort algorithm.**

**Code:**

static void Main(string[] args)

{

Console.WriteLine("How many values you want to add");

int n = Convert.ToInt32(Console.ReadLine());

string[] data = new string[n];

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

{

Console.WriteLine("Enter value");

data[i] = Console.ReadLine();

}

Console.WriteLine();

Console.WriteLine("Array before sorting");

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

{

Console.WriteLine(data[i]+" ");

}

Console.WriteLine();

MergeSortIterative(data);

Console.WriteLine("Array after sorting");

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

{

Console.WriteLine(data[i]+" ");

}

Console.WriteLine();

}

public static void MergeSortIterative(string[] C)

{

int i;

int LB;

for (i = 1; i <= C.Length - 1; i = 2 \* i)

{

for (LB = 0; LB < C.Length - 1; LB += 2 \* i)

{

int mid = LB + i - 1;

int UB = Math.Min(LB + 2 \* i - 1, C.Length - 1);

Merge(C, LB, mid, UB);

}

}

}

public static void Merge(string[] C, int LB, int mid, int UB)

{

int limit1 = mid-LB+1;

int limit2 = UB - mid;

string[] A = new string[limit1];

string[] B = new string[limit2];

for (int a = 0; a < limit1; a++)

{

A[a] = C[LB + a];

}

for (int b = 0; b < limit2; b++)

{

B[b] = C[mid + 1 + b];

}

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

for(;i<A.Length && j<B.Length; k++)

{

char[] arr1 = A[i].ToCharArray();

char[] arr2=B[j].ToCharArray();

if (arr1[0] <= arr2[0])

{

C[k] = A[i];

i++;

}

else

{

C[k] = B[j];

j++;

}

}

for (; i< A.Length; i++,k++)

{

C[k] = A[i];

}

for (; j < B.Length; j++)

{

C[k] = B[j];

}

}

**Output:**

