**LAB TASK**

**Data structure**

Arooj fatima

Roll no : 11

**Merge Sort**

#include <iostream>

Using namespace std;

Void merge(int arr[], int left, int mid, int right) {

Int n1 = mid – left + 1;

Int n2 = right – mid;

Int L[n1], R[n2];

For (int i = 0; i < n1; i++)

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

For (int i = 0; i < n2; i++)

R[i] = arr[mid + 1 + i];

Int i = 0, j = 0, k = left;

While (i < n1 && j < n2) {

If (L[i] ≤ R[j]) {

Arr[k] = L[i];

I++;

} else {

Arr[k] = R[j];

J++;

}

K++;

}

While (i < n1) {

Arr[k] = L[i];

I++;

K++;

}

While (j < n2) {

Arr[k] = R[j];

J++;

K++;

}

}

// Function to implement merge sort

Void mergeSort(int arr[], int left, int right) {

If (left < right) {

Int mid = left + (right – left) / 2;

// Sort first and second halves

mergeSort(arr, left, mid);

mergeSort(arr, mid + 1, right);

// Merge the sorted halves

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;

}

Int main() {

Int arr[] = {38, 27, 43, 3, 9, 82, 10};

Int n = sizeof(arr) / sizeof(arr[0]);

Cout ≪ “Original array: “;

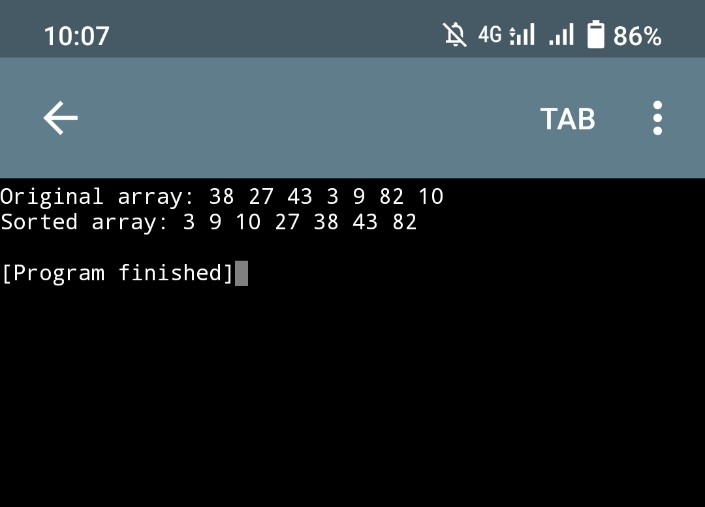
printArray(arr, n);

mergeSort(arr, 0, n – 1);

cout ≪ “Sorted array: “;

printArray(arr, n);

return 0;

}