#include <bits/stdc++.h>

using namespace std;

void swapIfGreater(long long arr1[], long long arr2[], int ind1, int ind2) {

if (arr1[ind1] > arr2[ind2]) {

swap(arr1[ind1], arr2[ind2]);

}

}

void merge(long long arr1[], long long arr2[], int n, int m) {

// len of the imaginary single array:

int len = n + m;

// Initial gap:

int gap = (len / 2) + (len % 2);

while (gap > 0) {

// Place 2 pointers:

int left = 0;

int right = left + gap;

while (right < len) {

// case 1: left in arr1[]

//and right in arr2[]:

if (left < n && right >= n) {

swapIfGreater(arr1, arr2, left, right - n);

}

// case 2: both pointers in arr2[]:

else if (left >= n) {

swapIfGreater(arr2, arr2, left - n, right - n);

}

// case 3: both pointers in arr1[]:

else {

swapIfGreater(arr1, arr1, left, right);

}

left++, right++;

}

// break if iteration gap=1 is completed:

if (gap == 1) break;

// Otherwise, calculate new gap:

gap = (gap / 2) + (gap % 2);

}

}

int main()

{

long long arr1[] = {1, 4, 8, 10};

long long arr2[] = {2, 3, 9};

int n = 4, m = 3;

merge(arr1, arr2, n, m);

cout << "The merged arrays are: " << "\n";

cout << "arr1[] = ";

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

cout << arr1[i] << " ";

}

cout << "\narr2[] = ";

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

cout << arr2[i] << " ";

}

cout << endl;

return 0;

}