Q1: Merge two arrays by satisfying given constraints

Given two sorted arrays X[] and Y[] of size m and n each where m >= n and X[] has exactly n vacant cells,

merge elements of Y[] in their correct position in array X[], i.e., merge (X, Y) by keeping the sorted order.

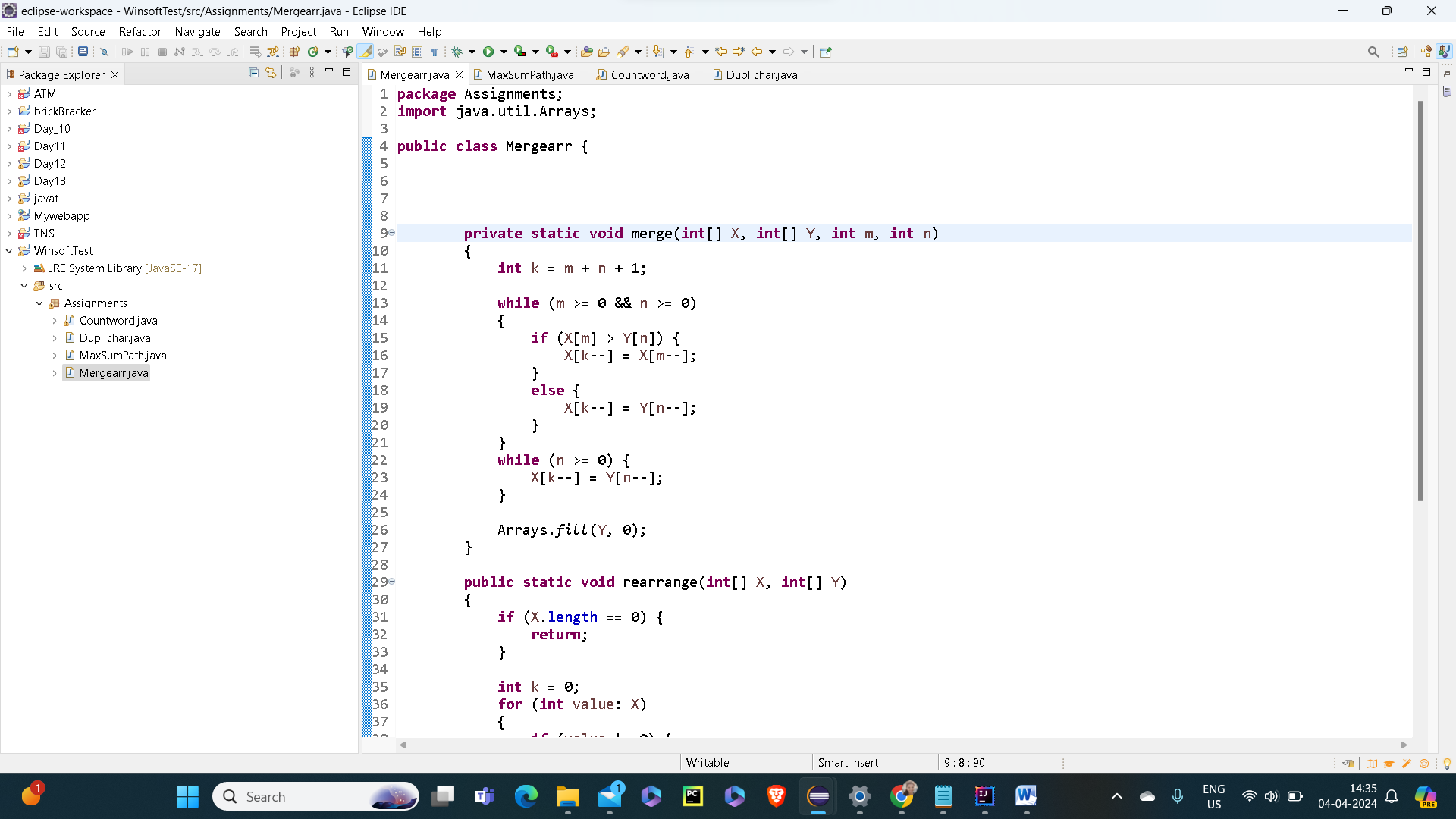
For example,

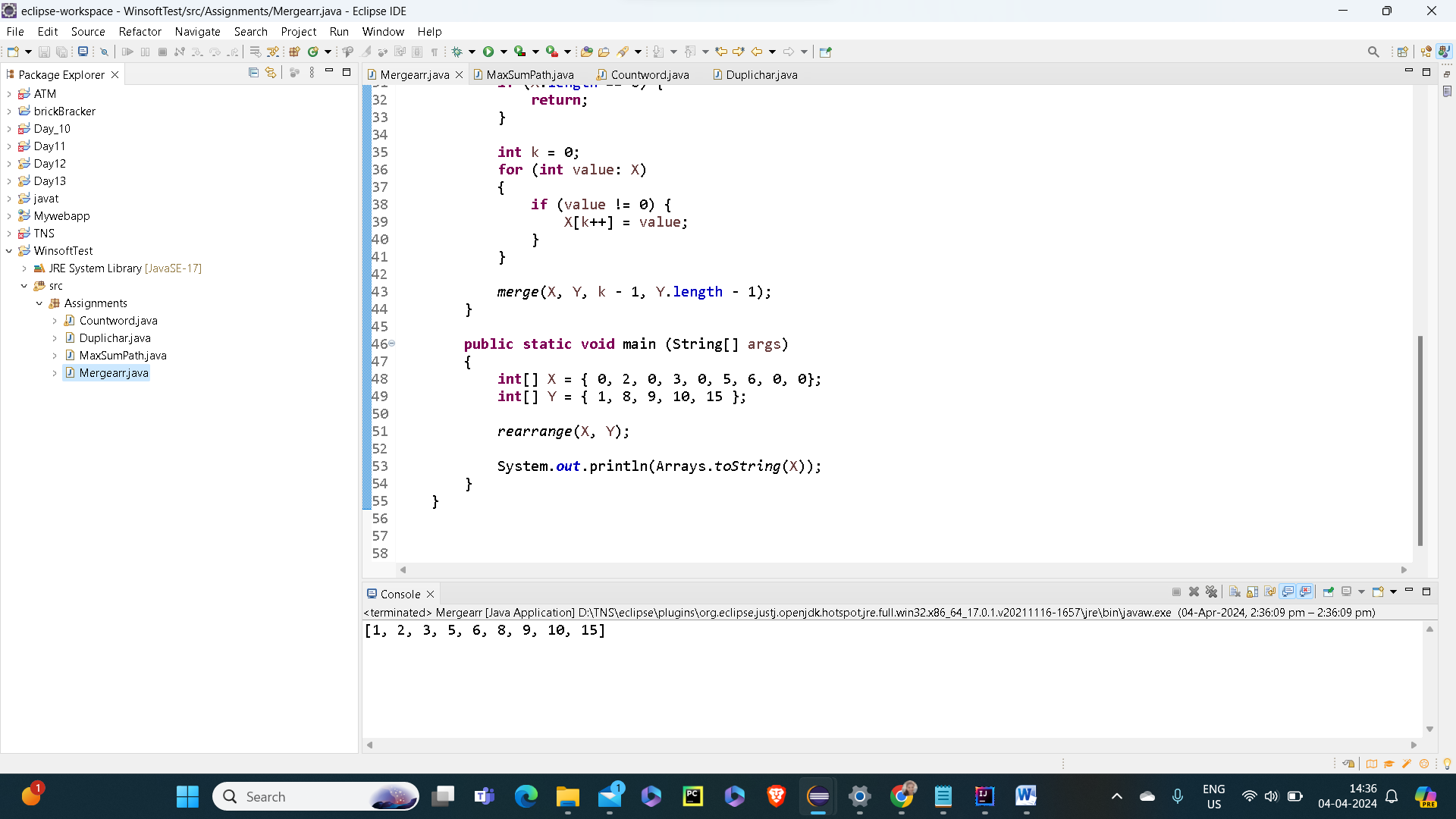
Input: X[] = { 0, 2, 0, 3, 0, 5, 6, 0, 0 }

Y[] = { 1, 8, 9, 10, 15 } The vacant cells in X[] is represented by 0

Output: X[] = { 1, 2, 3, 5, 6, 8, 9, 10, 15 }

Code:





Q2:Find maximum sum path involving elements of given arrays

Given two sorted arrays of integers, find a maximum sum path involving elements of both arrays whose sum is maximum.

We can start from either array, but we can switch between arrays only through its common elements.

For example,

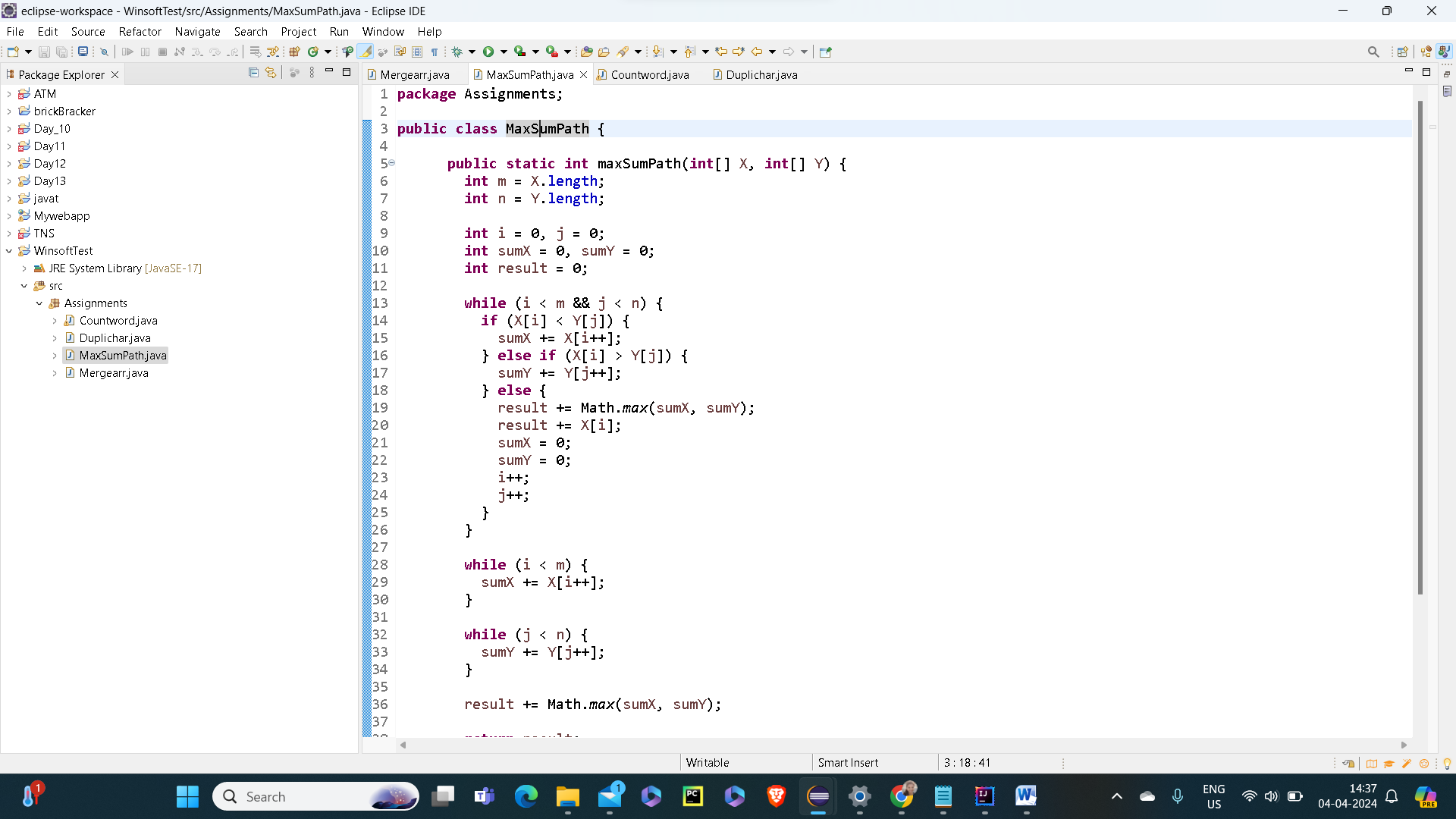
Input: X = { 3, 6, 7, 8, 10, 12, 15, 18, 100 }

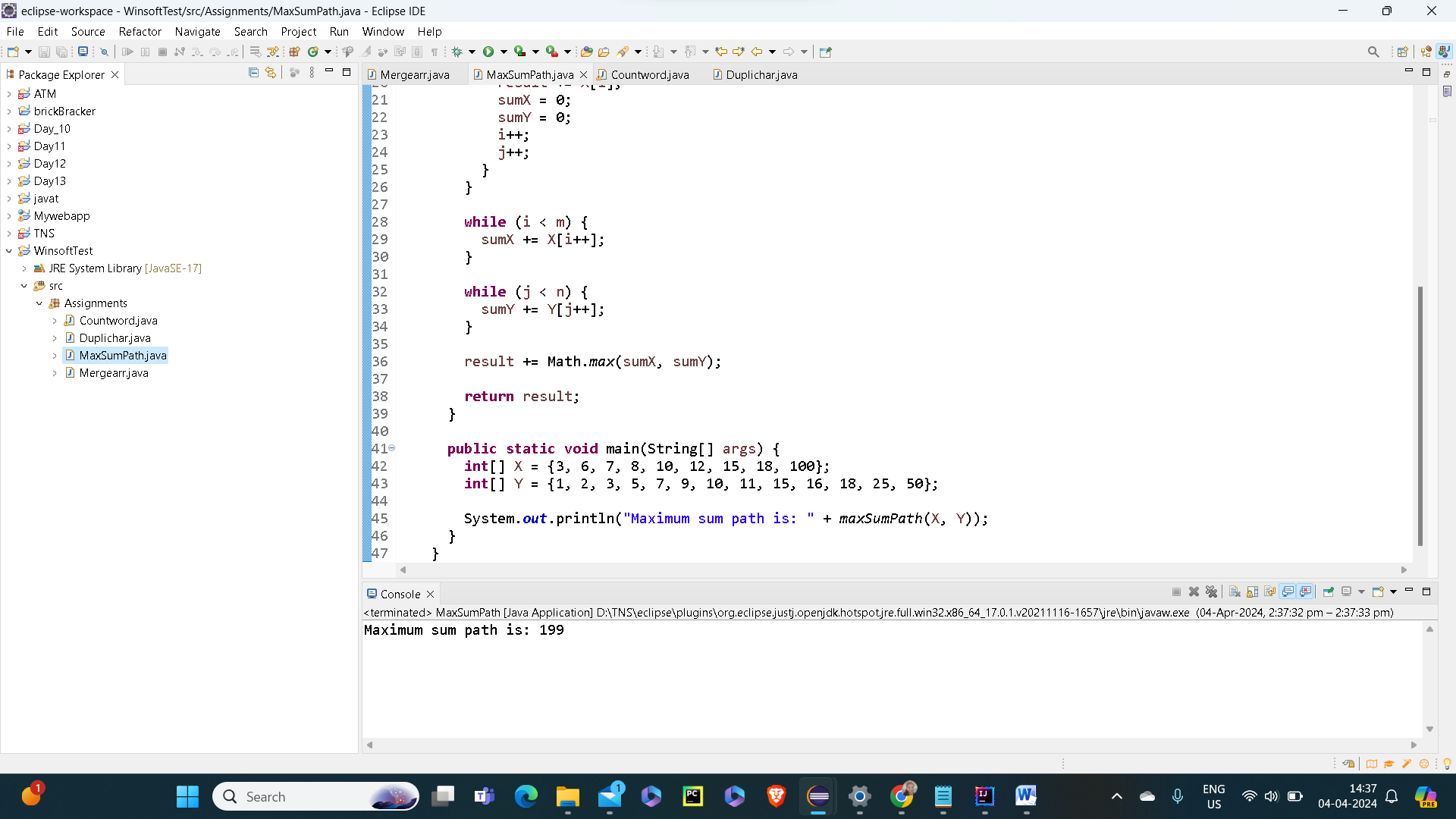
Y = { 1, 2, 3, 5, 7, 9, 10, 11, 15, 16, 18, 25, 50 }

The maximum sum path is: 1 —> 2 —> 3 —> 6 —> 7 —> 9 —> 10 —> 12 —> 15 —> 16 —> 18 —> 100

The maximum sum is 199

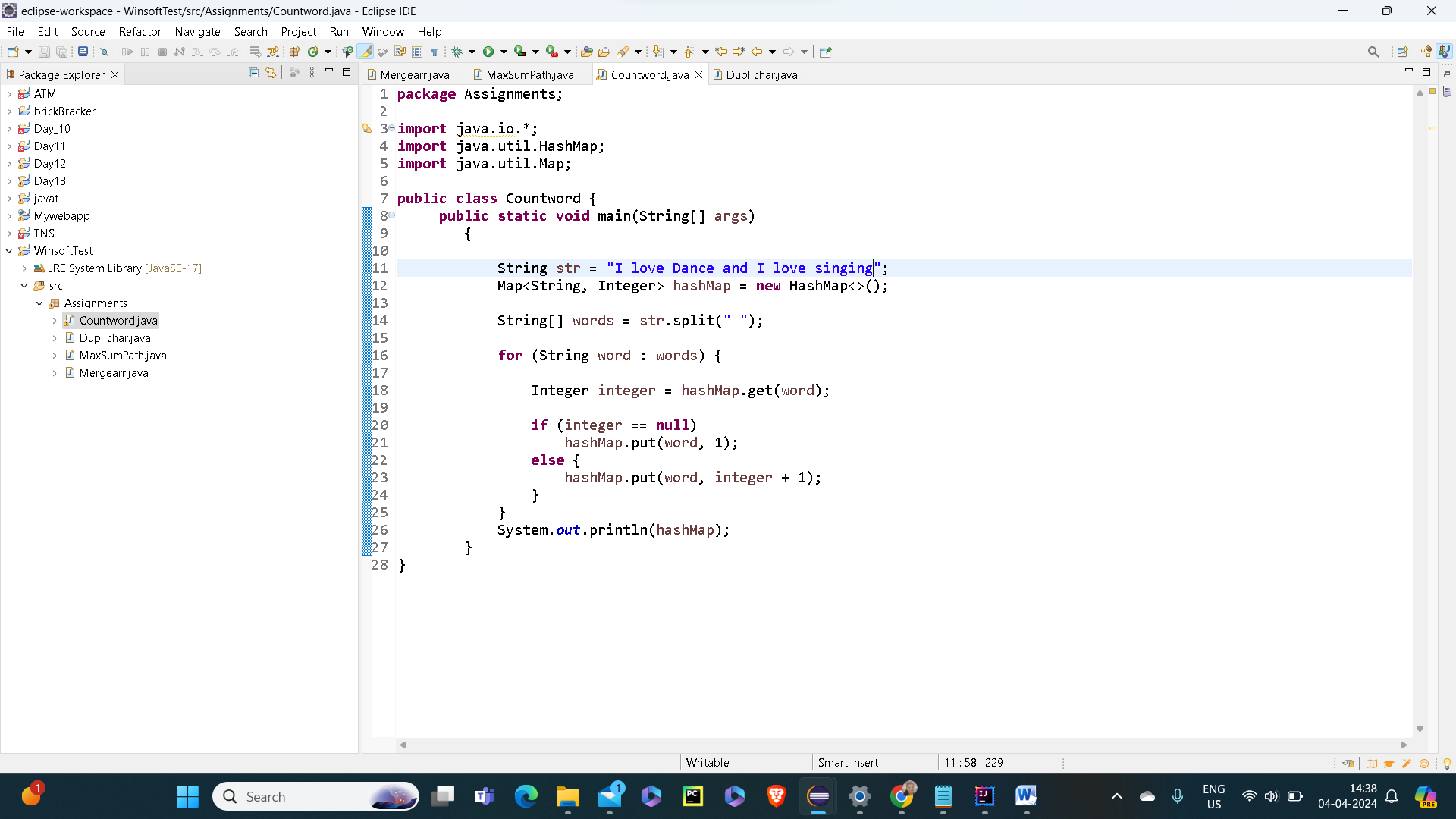
Code:

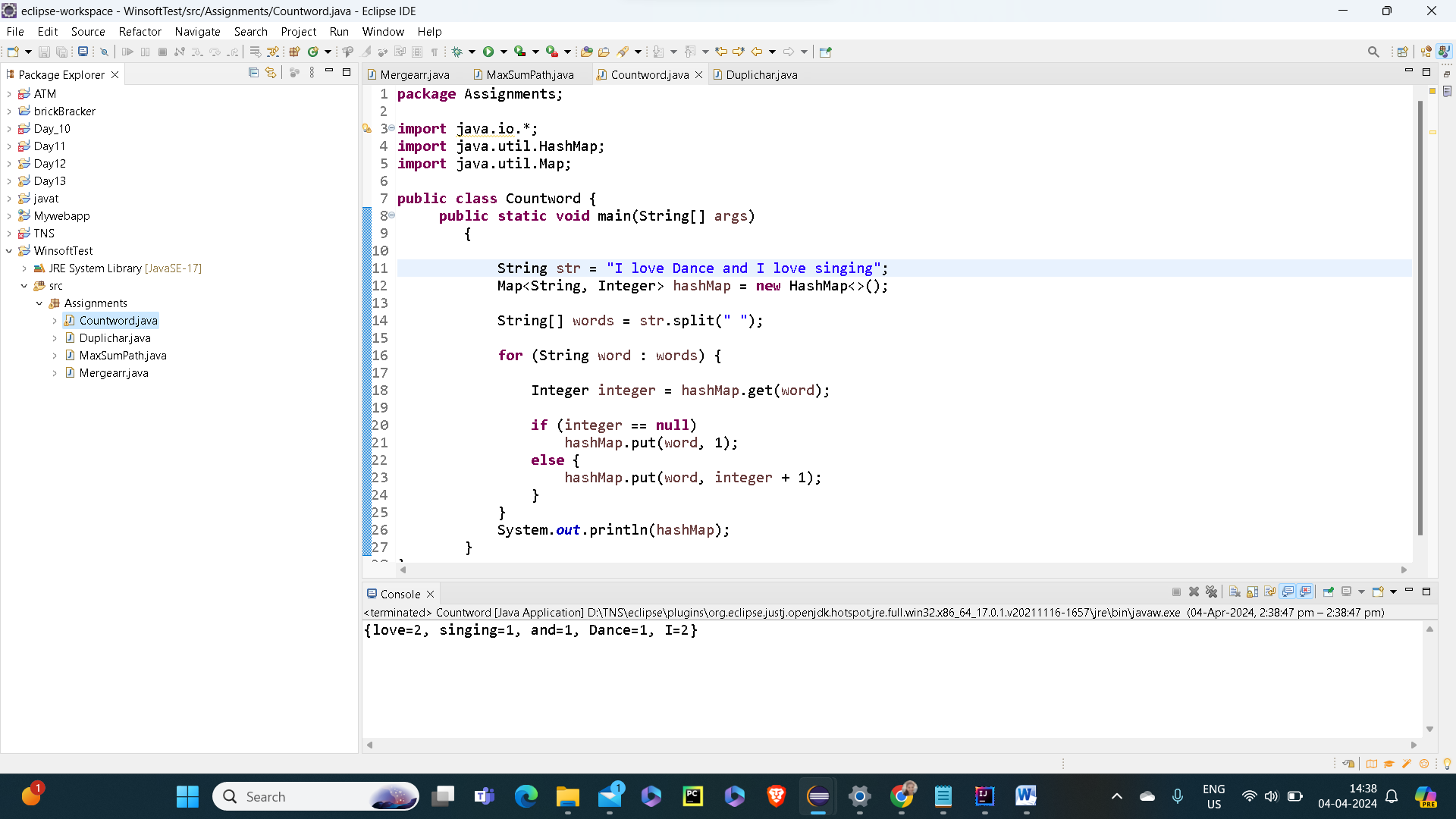




Q3:Write a Java Program to count the number of words in a string using HashMap.

Code:





Q4:Write a Java Program to find the duplicate characters in a string.

Code:

