CS211 Lab 2

MERGE SORT

You are given two strings **word1** and **word2**. Merge the strings by adding letters in alternating order, starting with word1. If a string is longer than the other, append the additional letters onto the end of the merged string.

Return the merged string.

Example 1:

Input: word1 = "abc", word2 = "pqr"

Output: "apbqcr"

Explanation: The merged string will be merged as so:

word1: a   b   c

word2:   p   q   r

merged: a p b q c r

Example 2:

Input: word1 = "ab", word2 = "pqrs"

Output: "apbqrs"

Explanation: Notice that as word2 is longer, "rs" is appended to the end.

```
word1:  a   b
word2:    p   q   r   s
merged: a p b q   r   s
```

Example 3:

Input: word1 = "abcd", word2 = "pq"

Output: "apbqcd"

Explanation: Notice that as word1 is longer, "cd" is appended to the end.

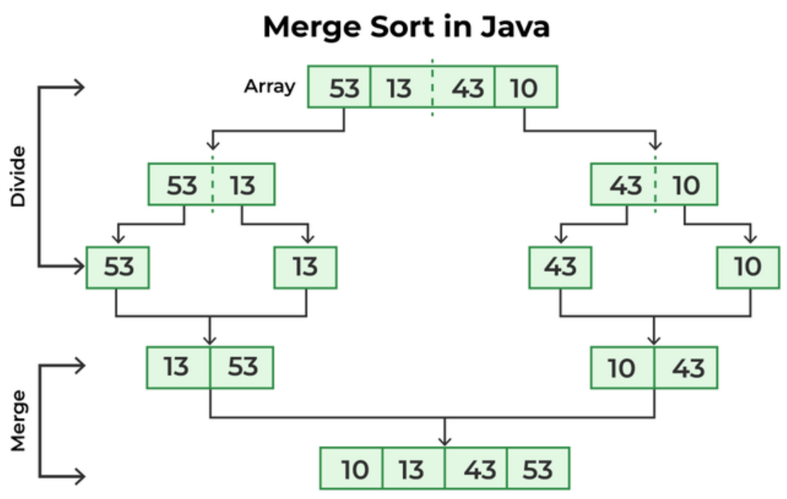
```
word1:  a   b   c   d
word2:    p   q
merged: a p b q c   d
```

Constraints:

1 <= word1.length, word2.length <= 100

word1 and word2 consist of lowercase English letters.

PART II



Create a new version of your program that assumes that **word1** and **word2** have their characters already sorted alphabetically. Your program should combine them in a such a way that the merged output String also has the property of being alphabetically sorted.

Input: word1 = "abcdhij", word2 = "efg"

Output: "abcdefghij"

Explanation: The “abcd” is added to the output first, then “efg”, then the “hij”

```
word1:  a   b   c   d            h  i  j
word2:                      e  f  g
merged: a   b   c   d  e  f  g  h  i  j
```

Congratulations, you’ve just created mergesort! The only part missing is a recursive method that breaks an input String into two halves and calls itself on both halves, then uses your merge method to stick both halves back together, with the base case being a single character. But creating that is only optional.

PEN AND PAPER EXERCISE

Show clearly using pen and paper how the output of the following program is computed – show all workings:

```
public class BitManipulation{
     public static void main(String[] args){
         System.out.println((( 43 & 27 ) << 3 ) | ( 14 | 9 ));
     }
}
```