

# LeetCode 75

Joel Castillo Espinosa \*

## Contents

<b>1768 Merge Strings Alternately</b>	<b>1</b>
DESCRIPTION . . . . .	1
SOLUTION . . . . .	2

## 1768 Merge Strings Alternately <sup>1</sup>

### DESCRIPTION

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.

### Examples

- Example 1:
  - Input: word1 = “abc”, word2 = “pqr”
  - Output: “apbqcr”
- Example 2:
  - Input: word1 = “ab”, word2 = “pqrs”
  - Output: “apbqrs”
- Example 3:
  - Input: word1 = “abcd”, word2 = “pq”
  - Output: “apbqcd”

### Constraints:

- $1 \leq \text{word1.length}, \text{word2.length} \leq 100$
- **word1** and **word2** consist of lowercase English letters

---

\*Email: [jocastillo@colmex.mx](mailto:jocastillo@colmex.mx). For more content visit my website: <https://joelcastillo.netlify.app> If you have any questions or suggestions, I'd be grateful to receive your message.

<sup>1</sup>This problem is originally from Leetdode, you can find it in [Leetcode](#).

## SOLUTION <sup>2</sup>

```
merg_str <- function(word1, word2){  
  # split the words in letters  
  split1 <-strsplit(word1,"")  
  split2 <-strsplit(word2,"")  
  
  l_w1 <- length(split1[[1]])  
  l_w2 <- length(split2[[1]])  
  
  # condition  
  if(l_w1 == 0 | l_w2== 0) {warning("Length of world1 and world2 have to be > 0 ")}  
  
  else {  
    string <- "" # element that contains the merge  
  
    # in each interaction, take an element (letter or "") per word  
    for (i in 1:max(l_w1,l_w2)) { # to include all elements  
      if (is.na(split1[[1]][i])) {letter1 <- ""}  
      else{ letter1 <-split1[[1]][i] }  
  
      if (is.na(split2[[1]][i])) {letter2 <- ""}  
      else { letter2 <- split2[[1]][i] }  
  
      string <- paste(string,letter1,letter2, sep = "")  
    }  
    return(string)  
  }  
}
```

### Examples using the function

We can use the examples presented before.

```
merg_str("abc","pqr")
```

```
## [1] "apbqcr"
```

```
merg_str(word1 = "ab", word2 = "pqrs")
```

```
## [1] "apbqrs"
```

In addition, we can also check the restriction in the length of the words.

```
merg_str("", "Hello")
```

```
## Warning in merg_str("", "Hello"): Length of world1 and world2 have to be > 0
```

---

<sup>2</sup>This solution is entirely my authorship. I used R version 4.4.1 (2024-06-14 ucrt).