

## Question - 1

### Merge Strings

You are merging data from two sources connected to a network access point to create a new data packet.

You must merge strings *a* and *b*, and then return a single merged string. A merge operation on two strings is described as follows:

Append alternating characters from *a* and *b*, respectively, to some new string, *mergedString*.

Once all of the characters in one of the strings have been merged, append the remaining characters in the other string to *mergedString*.

As an example, assume you have two strings to merge: 'abc' and 'stuvwx'. Alternate between the first and second strings as long as you can:

'a' + 's' + 'b' + 't' + 'c' + 'u'. At this point you have traversed the first string and have generated 'asbtcu'. The remainder of the second string, 'vwx' is now added to the end of the string, creating 'asbtcuwx'.

### Function Description

Complete the function `mergeStrings` in the editor below. The function must return the merged string.

`mergeStrings` has the following parameter(s): *a*: first string

*b*: second string

### Constraints

$1 \leq |a|, |b| \leq 25000$

### Input Format For Custom Testing Sample Case 0

#### Sample Input 0

```
abc def
```

#### Sample Output 0

```
adbecf
```

#### Explanation 0

*a* = abc *b* = def

Taking alternate characters from both the strings, we get adbecf

### Sample Case 1

## Question - 2

### Distinct Pairs

In this challenge, you will be given an array of integers and a target value. Determine the number of distinct pairs of elements in the array that sum to the target value. Two pairs (*a*, *b*) and (*c*, *d*) are considered to be distinct if and only if the values in sorted order do not match, i.e., (1, 9) and (9, 1) are indistinct but (1, 9) and (9, 2) are distinct.

For instance, given the array [1, 2, 3, 6, 7, 8, 9, 1], and a target value of 10, the seven pairs (1,9), (2,8), (3,7), (8, 2), (9, 1), (9, 1), and (1, 9) all sum to 10 and only three distinct pairs: (1, 9), (2, 8), and (3, 7).

### Function Description

Complete the function `numberOfPairs` in the editor below. The function must return an integer, the total number of distinct pairs of elements in the array that sum to the target value.

`numberOfPairs` has the following parameter(s): `a[a[0],...a[n-1]]`: an array of integers to select pairs from `k`: target integer value to sum to

### Constraints

$$5 \leq n \leq 5 \times 10$$

$$0 \leq a[i] \leq 10$$

$$0 \leq k \leq 5 \times 10$$

### Input Format for Custom Testing

#### Sample Case 0

#### Sample Input 0

6 1 3 46 1 3 9 47

#### Sample Output 0

1

#### Explanation 0

`a = [1, 3, 46, 1, 3, 9]`, `k = 47`

There are 4 pairs of unique elements where `a[i] + a[j] = k`:

1. (`a[0] = 1`, `a[2] = 46`) 2. (`a[2] = 46`, `a[0] = 1`) 3. (`a[2] = 46`, `a[3] = 1`) 4. (`a[3] = 1`, `a[2] = 46`)

In the list above, all four pairs contain the same values. We only have 1 distinct pair, (1, 46).