Common Prefix Length

Given a string, split the string into two substrings at every possible point. The rightmost substring is a *suffix.* The beginning of the string is the *prefix.* Determine the lengths of the common prefix between each suffix and the original string. Sum and return the lengths of the common prefixes.

Example

Consider the only string in the array *inputs = ['abcabcd']*. Each suffix is compared to the original string.

Remove to leave suffix	**Suffix**	**Common Prefix**	**Length**
''	'abcabcd'	'abcabcd'	7
'a'	'bcabcd'	''	0
'ab'	'cabcd'	''	0
'abc'	'abcd'	'abc'	3
'abca'	'bcd'	''	0
'abcab'	'cd'	''	0
'abcabc'	'd'	''	0

The sum is *7 + 0 + 0 + 3 + 0 + 0 + 0 = 10.*

Function Description

Complete the function *commonPrefix* in the editor below.

commonPrefix has the following parameter(s):

string inputs[n]: an array of strings

Returns:

int[]: the sums of the common prefix lengths for each test case

Constraints

$1 \le n \le 10$

$1 \le | inputs[i] | \le 10^5$

Each *inputs[i]* contains only letters in the range *ascii[a-z]*.

Input Format For Custom Testing

The first line contains the number of test cases *n*.
Each of the next *n* lines contains a string, *inputs[i],* one for each test case.

Sample Case 0

Sample Input

```
STDIN     Function
-----     -----
1      → number of test cases n = 1
ababaa → inputs = ['ababaa']
```

Sample Output

11

Explanation

n = 1

inputs = ['ababaa']
The suffixes are *['ababaa', 'babaa', 'abaa', 'baa', 'aa', 'a'].* The common prefix lengths of each of these suffixes with the original string are *[6, 0, 3, 0, 1, 1]* respectively, and they sum to *11*.

Sample Case 1

Sample Input

```
STDIN     Function
-----     -----
1      →  number of test cases n = 1
aa     →  inputs = ['aa']
```

Sample Output

```
3
```

Explanation

n = 1

inputs = ['aa']

The suffixes are *['aa', 'a'].* The common prefix lengths of each of these suffixes with the original string are *[2, 1]* which sum to *3.*