Problem 3271: Hash Divided String

Problem Information

Difficulty: Medium
Acceptance Rate: 83.09%
Paid Only: No
Tags: String, Simulation

Problem Description

You are given a string `s` of length `n` and an integer `k`, where `n` is a **multiple** of `k`. Your task is to hash the string `s` into a new string called `result`, which has a length of `n / k`.

First, divide `s` into `n / k` **substrings** , each with a length of `k`. Then, initialize `result` as an **empty** string.

For each **substring** in order from the beginning:

* The **hash value** of a character is the index of that character in the **English alphabet** (e.g., `'a' -> 0`, `'b' -> 1`, ..., `'z' -> 25`). * Calculate the _sum_ of all the **hash values** of the characters in the substring. * Find the remainder of this sum when divided by 26, which is called `hashedChar`. * Identify the character in the English lowercase alphabet that corresponds to `hashedChar`. * Append that character to the end of `result`.

Return `result`.

Example 1:

Input: s = "abcd", k = 2

Output: "bf"

Explanation:

First substring: `"ab"`, `0 + 1 = 1`, `1 % 26 = 1`, `result[0] = 'b'`.

Second substring: `"cd"`, `2 + 3 = 5`, `5 % 26 = 5`, `result[1] = 'f'`.

Example 2:

Input: s = "mxz", k = 3

Output: "i"

Explanation:

The only substring: `"mxz"`, `12 + 23 + 25 = 60`, `60 % 26 = 8`, `result[0] = 'i'`.

Constraints:

* `1 <= k <= 100` * `k <= s.length <= 1000` * `s.length` is divisible by `k`. * `s` consists only of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
string stringHash(string s, int k) {

}
};
```

Java:

```
class Solution {
public String stringHash(String s, int k) {

}
}
```

Python3:

```
class Solution:
def stringHash(self, s: str, k: int) -> str:
```