

# Problem 1933: Check if String Is Decomposable Into Value-Equal Substrings

## Problem Information

**Difficulty:** Easy

**Acceptance Rate:** 50.76%

**Paid Only:** Yes

**Tags:** String

## Problem Description

A \*\*value-equal\*\* string is a string where \*\*all\*\* characters are the same.

\* For example, `"1111"` and `"33"` are value-equal strings. \* In contrast, `"123"` is not a value-equal string.

Given a digit string `s`, decompose the string into some number of \*\*consecutive value-equal\*\* substrings where \*\*exactly one\*\* substring has a \*\*length of\*\* `2` and the remaining substrings have a \*\*length of\*\* `3` .

Return `true` \_if you can decompose\_ `s` \_according to the above rules. Otherwise, return\_ `false` .

A \*\*substring\*\* is a contiguous sequence of characters in a string.

**Example 1:**

**Input:** s = "000111000" **Output:** false **Explanation:** s cannot be decomposed according to the rules because ["000", "111", "000"] does not have a substring of length 2.

**Example 2:**

**Input:** s = "00011111222" **Output:** true **Explanation:** s can be decomposed into ["000", "111", "11", "222"].

**\*\*Example 3:\*\***

**\*\*Input:\*\*** s = "011100022233" **\*\*Output:\*\*** false **\*\*Explanation:\*\*** s cannot be decomposed according to the rules because of the first '0'.

**\*\*Constraints:\*\***

\* `1 <= s.length <= 1000` \* `s` consists of only digits '0' through '9'.

## Code Snippets

**C++:**

```
class Solution {  
public:  
    bool isDecomposable(string s) {  
        }  
    };
```

**Java:**

```
class Solution {  
public boolean isDecomposable(String s) {  
    }  
}
```

**Python3:**

```
class Solution:  
    def isDecomposable(self, s: str) -> bool:
```