# Problem 906: Super Palindromes

## Problem Information

**Difficulty:** Hard
**Acceptance Rate:** 39.79%
**Paid Only:** No
**Tags:** Math, String, Enumeration

## Problem Description

Let's say a positive integer is a **super-palindrome** if it is a palindrome, and it is also the square of a palindrome.

Given two positive integers `left` and `right` represented as strings, return _the number of**super-palindromes** integers in the inclusive range_ `[left, right]`.

**Example 1:**

**Input:** left = "4", right = "1000" **Output:** 4 **Explanation** : 4, 9, 121, and 484 are superpalindromes. Note that 676 is not a superpalindrome: 26 * 26 = 676, but 26 is not a palindrome.

**Example 2:**

**Input:** left = "1", right = "2" **Output:** 1

**Constraints:**

* `1 <= left.length, right.length <= 18` * `left` and `right` consist of only digits. * `left` and `right` cannot have leading zeros. * `left` and `right` represent integers in the range `[1, 1018 - 1]`. * `left` is less than or equal to `right`.

## Code Snippets

### C++:

```
class Solution {
public:
int superpalindromesInRange(string left, string right) {

}
};
```

### Java:

```
class Solution {
public int superpalindromesInRange(String left, String right) {

}
}
```

### Python3:

```
class Solution:
def superpalindromesInRange(self, left: str, right: str) -> int:
```