

Problem 2843: Count Symmetric Integers

Problem Information

Difficulty: [Easy](#)

Acceptance Rate: 83.09%

Paid Only: No

Tags: Math, Enumeration

Problem Description

You are given two positive integers `low` and `high`.

An integer `x` consisting of `2 * n` digits is **symmetric** if the sum of the first `n` digits of `x` is equal to the sum of the last `n` digits of `x`. Numbers with an odd number of digits are never symmetric.

Return the number of symmetric integers in the range `[low, high]`.

Example 1:

Input: `low = 1, high = 100` **Output:** `9` **Explanation:** There are 9 symmetric integers between 1 and 100: 11, 22, 33, 44, 55, 66, 77, 88, and 99.

Example 2:

Input: `low = 1200, high = 1230` **Output:** `4` **Explanation:** There are 4 symmetric integers between 1200 and 1230: 1203, 1212, 1221, and 1230.

Constraints:

`1 <= low <= high <= 104`

Code Snippets

C++:

```
class Solution {  
public:  
    int countSymmetricIntegers(int low, int high) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int countSymmetricIntegers(int low, int high) {  
  
    }  
}
```

Python3:

```
class Solution:  
    def countSymmetricIntegers(self, low: int, high: int) -> int:
```