Leetcode Problem 1. (Easy)

Roman to Integer

Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.

**Symbol** **Value**

I 1

V 5

X 10

L 50

C 100

D 500

M 1000

For example, 2 is written as II in Roman numeral, just two ones added together. 12 is written as XII, which is simply X + II. The number 27 is written as XXVII, which is XX + V + II.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as IX. There are six instances where subtraction is used:

* I can be placed before V (5) and X (10) to make 4 and 9.
* X can be placed before L (50) and C (100) to make 40 and 90.
* C can be placed before D (500) and M (1000) to make 400 and 900.

Given a roman numeral, convert it to an integer.

**Example 1:**

**Input:** s = "III"

**Output:** 3

**Explanation:** III = 3.

**Example 2:**

**Input:** s = "LVIII"

**Output:** 58

**Explanation:** L = 50, V= 5, III = 3.

**Example 3:**

**Input:** s = "MCMXCIV"

**Output:** 1994

**Explanation:** M = 1000, CM = 900, XC = 90 and IV = 4.

**Constraints:**

* 1 <= s.length <= 15
* s contains only the characters ('I', 'V', 'X', 'L', 'C', 'D', 'M').
* It is **guaranteed** that s is a valid roman numeral in the range [1, 3999].

Link : <https://leetcode.com/problems/roman-to-integer/>

int result = 0;

int prevValue = 0; // to keep track of the previous numeral value

for (int i = s.length() - 1; i >= 0; i--) { // iterate from right to left

int currValue = getRomanValue(s.charAt(i));

if (currValue < prevValue) { // if current value is less than previous value, subtract it

result -= currValue;

} else { // otherwise, add it to the result

result += currValue;

}

prevValue = currValue; // update previous value

}

return result;

}

private int getRomanValue(char c) { // helper function to get the value of a Roman numeral

switch (c) {

case 'I':

return 1;

case 'V':

return 5;

case 'X':

return 10;

case 'L':

return 50;

case 'C':

return 100;

case 'D':

return 500;

case 'M':

return 1000;

default:

return 0;

}

}

}

