



[CS 11] Prac 6h – Ramon Numerals

Problem Statement

Roman numerals have this rule that if a letter appears to the left of a higher-valued letter, then its value is subtracted instead of added. For example, IV = 5 - 1 = 4 while VI = 5 + 1 = 6.

Ramon is having none of that. He thinks it's overcomplicated. So, he invented his own numeral system, which we can call **Ramon numerals**. Ramon numerals are similar to Roman Numerals, but simpler—one simply adds up the values of all the letters. For example, IV and VI are two ways of writing the number six.

We can also look at **generalized Ramon numerals**, where someone selects letters along with their numeric values. For example, if Y = 10, A = 3, and P = 5, then "PAPAYA" is 29.

Given a Ramon numeral, what is its value?

[Submit solution](#)

[CS 11]

Practice 6

[My submissions](#)

✓ Points: 125 (partial!)

⌚ Time limit: 6.0s

💻 Memory limit: 1G

Author:

kvatienza (Kevin Atienza)

Problem type

✓ Allowed languages

NONE, py3

Task Details

Your task is to implement a function called `ramon_numeral_value`. This function has two parameters:

- the first parameter is a `dict` representing the value of each letter in a generalized Ramon numeral system.
- the second parameter is a `str` representing a Ramon numeral.

The function must return an `int` denoting the value of the Ramon numeral of the given matrix.

Restrictions

(See 6a for more restrictions)

For this problem:

- Loops and lists are allowed.
- Up to 1 function definition is allowed.
- Recursion is **disallowed**. (The recursion limit has been greatly reduced.)
- Sets and dictionaries are allowed.
- Generators and comprehensions are allowed.
- The following names are allowed: `set`, `dict`, `iter`, `next`, `any`, `all`, `popitem`, `setdefault`, `update`, `add`, `discard`.
- The source code limit is 100.

Example Calls

Example 1 Function Call

```
ramon_numeral_value({
```

```
    'I': 1,  
    'V': 5,  
    'X': 10,  
    'L': 50,  
    'C': 100,  
    'D': 500,  
    'M': 1000,
```

```
}, 'VIII')
```

[Copy](#)

Example 1 Return Value

```
8
```

[Copy](#)

Example 2 Function Call

```
ramon_numeral_value({
```

```
    'I': 1,  
    'V': 5,  
    'X': 10,  
    'L': 50,  
    'C': 100,  
    'D': 500,  
    'M': 1000,
```

```
}, 'IV')
```

[Copy](#)

Example 2 Return Value

```
6
```

[Copy](#)

Constraints

- The function `ramon_numeral_value` will be called at most 100 times.
- Every Ramon numeral passed as second argument has length between 1 and 2,000 and will be *valid*— all its letters will appear as keys in the `dict`.
- The `dict`'s keys are all uppercase letters.
- The value of each letter is an integer between 1 and 10,000.

Scoring

- You get 125 ❤ points if you solve all test cases.

Clarifications

[Report an issue](#)

No clarifications have been made at this time.