|  |
| --- |
| **We call it the PEDAC process:** |
|  |
| Understand the Problem |
| Examples / Test Cases |
| Data Structure and Algorithms |
| Code |

Roman

Convert a number into a roman numeral string

Test numbers 1-3000

Input is a number and output is a string representing the number. We can create a lookup table that corresponds the number to a Roman numeral. As we divide the number by lookup number we can record that Roman numeral. If the remainder is larger than the lookup number then do it again, otherwise move on to the smallest number. Stop if remainder is 0.

Word Count

-Split phrase into separate words, ignoring everything else around the words, except apostrophe in words like “you’re”, “I’ll”, “didn’t”

-Convert the words all to lowercase, so words are qualified as the same despite capitalization, DOG, Dog, dog, should all belong to the same category.

- Count an instance of each word

Test Cases

-empty string

-single word

-multiple word phrase

-full sentence

- Words and random characters mixed together

- sentence with quoted words

- sentence with floating numbers

Data Structure

Input is a string and output is an object containing the word as key and number of occurrences as value

We need to create regex to match just the needed characters to get an array of words. This is the toughest part of the entire challenge. Afterwards, looping through the array, need to add 1 to value of the key or if it’s not a part of the object then add the word as the key with a value of 1.

Next step would be to figure out floating numbers in the string,

\b[a-z'0-9]+\.\*[0-9]\*\b

Sum of Multiples

* Find multiples for a given number
* Parameters include: Set of multiples, and limit to find multiples
* Sum multiples, which have to be unique. If they are overlapping like 15 is a multiple of 5 and 3, then discard the extra multiple.
* If no argument is given for multiples, then default to 3 and 5
* If no arguments given at all, raise an error
* If argument is not a number, raise an error

Test Cases

-Arguments which aren’t numbers, all or none

-No arguments

-Limit is 0

-Multiple is 0

-1 argument

-2 arguments

-3 arguments

Input is a set of numbers. Can be one or more. The number of multiples is limit/multiple, can go through a loop up to but not including maximum number and add multiples to an array. This can be done for each multiple. Each multiple will have its own array which can be concatenated. Once we have a one big array of all the multiples we can make a new array of unique numbers. Then loop through the array of unique multiples and add all the numbers up to one sum. Return the sum.