romannumeral.py Flowchart

Convert integers from [1, 10_000] into roman numerals.

romannumeral(positive_int)

Start →
designate list of romannumeral_power10s: ["I", "X", "C", "M"] →
designate list of romannumeral_5power10s: ["V", "L", "D"]

Is the positive_number 10_000?

No →
Find the thousands_digit of the number. (0 means there is no thousands digit) →
Pass the digits up to hundreds to HUNDREDS_ROMANNUMERAL →
Concatenate "M" * thousands_digit + to the return of HUNDREDS_ROMANNUMERAL →
RETURN result

Yes →
result is "M" * 10 = MMMMMMMMMM →
RETURN result

HUNDREDS_ROMANNUMERAL(positive_number) →
Assign empty string to roman_number = "" →
Create a digits_iterative, a reverse-ordered list of the digits →
Look at each digit in the list starting from the first one to the last →
Take nth digit →
Translate this digit by passing it an its place_value to GET_ROMANNUMERAL →
Concatenate return to the front of the roman_number string →
Is this last digit in the list?

Is this last digit in the list? **No** → Proceed next digit in the list → Take nth digit

Is this last digit in the list? → RETURN roman_number

GET_ROMANNUMERAL(digit, place_value) →
Assign roman_1 letter by looking at romannumeral_power10s list based on the place_value →
Assign roman_1 letter by looking at romannumeral_5power10s list based on the place_value →
Is digit = 4?

Is digit = 4? **Yes** → RETURN roman_1 + roman_5

No →
Is digit = 9? **Yes** → digit = digit - 5 → RETURN roman_5 + roman_1 * digit

No →
Is digit [5, 8]? **Yes** → RETURN roman_1 * digit