

European Iteration.

- Prepared by: [Sagun Shakya \(https://github.com/sagsshakya\)](https://github.com/sagsshakya)
- MSc. Data Science
- GITAM Institute of Science, Visakhapatnam.
- Email: sags.shakya@gmail.com (<mailto:sags.shakya@gmail.com>)

We know about number systems: The Roman numerals and the alternative place-value system with a given base.

For the purposes of this problem, we limit ourselves to

Roman numerals with values up to 3999 MMMCMXCIX "Place value system" numbers having bases from 2 (with possible symbols 0, 1) through 36 (with possible symbols 0, 1, ..., 9, A, ... ,Z) Consider the following procedure:

Accept a natural number N (in base 10). If N lies in the closed interval [1,3999], i.e. between 1 and 3999 (both inclusive), convert N to R, its Roman numeral representation; else output N as the result and stop. Identify the base in which the value of R, now considered to be in "place value system", is least and calculate its value in base 10, replacing N with this value. Repeat from step 2. Constraints $1 \leq N \leq 3999$

Input Format A single Integer N. Output Converted N Test Case Example Input 1 Output 45338950 Explanation The procedure goes as follows in this case:

Accept N = 1. Since 1 lies in [1,3999], convert it to Roman R = I The least value of I (in bases 19 and above) is 18 in base 10. Hence N = 18. Repeating step 2, since 18 lies in [1,3999], convert it to R = XVIII The least value of XVIII (in base 34) is $3334^4 + 3134^3 + 1834^2 + 1834 + 18$ or N = 45338950 Repeating step 2, since 45338950 lies outside [1,3999] output 45338950 and stop.

```
In [30]: N = int(input())
```

2

```
In [31]: # Function to convert decimal to Roman Numerals
def decimalToRoman(decimal_number):
    num = [1, 4, 5, 9, 10, 40, 50, 90,
           100, 400, 500, 900, 1000]
    symbols = ["I", "IV", "V", "IX", "X", "XL",
               "L", "XC", "C", "CD", "D", "CM", "M"]
    i = 12
    total_roman = ''
    while decimal_number:
        quotient = decimal_number // num[i]
        decimal_number %= num[i]

        while quotient:
            total_roman += symbols[i]
            quotient -= 1
        i -= 1
    return total_roman
# This code is contributed by
# sanjeev2552
```

```
In [32]: while (N>=1) and (N<=3999):
    roman = decimalToRoman(N)
    # print(N)
    highest_place_value = sorted(list(set(roman)))[-1]
    # print(highest_place_value)
    d = int(roman, ord(highest_place_value) - 54)
    N = d
    # print('-----')
print(N)
```

```
2
I
-----
360
X
-----
16522299
```