

① 01h : 22m : 55s to test end



☆ Romanizer



Complete the function *romanizer*, that has one parameter, an array, *numbers*, of n integers. The function should return an array of n strings, such that i^{th} element in the array is the roman representation of i^{th} element of numbers.

_

You can use the roman representation of following integers:

3

4

5

Integer	Roman Representation
1	I
4	IV
5	V
9	IX
10	X
40	XL
50	L
90	XC
100	С
400	CD
500	D
900	СМ
1000	М

Input Format

The first line of the input is an integer n, total number of elements in the array, numbers. Each of the next n lines contains a single integer, denoting the elements of the array.

Constraints



① 01h : 22m : 55s to test end

Output Format

?

The function should return an array of n strings, such that i^{th} element in the array is the roman representation of i^{th} element of numbers.

Sample Input 1



2

3

4

5 1 2 3 4 5

5 Sample Output 1

I II IV V

Sample Input 2

Sample Output 2

LXXV XCIX C L

YOUR ANSWER



① 01h : 22m : 55s to test end

```
=
                                            Python 2
          Draft saved 08:21 pm
                             Original code
                                                                          Ö
8
         1
            #!/bin/python
         2
         3
            import sys
         4
            import os
         5
         6
3
         7
            # Complete the function below.
         8
         9
4
        10 ▼ def romanizer(numbers):
                 dic = \{\}
        11
5
        12
        13
                 dic
        14
                 ret = []
        15
        16
            def getRoman(dic, num)
        17
        18
        19
        20 ▶ f = open(os.environ['OUTPUT PATH'], 'w')↔
        22
        23
            _numbers cnt = 0
        24
            _numbers_cnt = int(raw_input())
        25
             numbers i=0
             numbers = []
        26
        27 ▼ while _numbers_i < _numbers_cnt:
                 numbers item = int(raw input());
        28
                 _numbers.append(_numbers_item)
        29
        30
                 numbers i+=1
        31
        32
        33
            res = romanizer( numbers);
        34
            for res cur in res:
        35
                 f.write( str(res_cur) + "\n" )
        36
        37
            f.close()
        38
                                                             Line: 12 Col: 5
```



O1h: 22m: 55s to test end



♣ Download sample test cases The input/output files have Unix line endings. Do not use Notepad to edit them on windows.





About Privacy Policy Terms of Service

2

3

4

5