

تحويل الأعداد الصحيحة إلى أرقام رومانية

الأرقام الرومانية تتكون من سبعة رموز I, V, X, L, C, Q D, M :

رمز	قيمة
`I`	1
`V`	5
`X`	10
`L`	50
`C`	100
`D`	500
`M`	1000

على سبيل المثال، 2 تُكتب ك II بالأرقام الرومانية، وهكذا.

الرقم 12 يُكتب ك XII وهو ما يعني $X + II$.

الرقم 27 يُكتب ك XXVII، وهو ما يعني $XX + V + II$

تُكتب الأرقام الرومانية عادةً من الأكبر إلى الأصغر من اليسار إلى اليمين. ومع ذلك، الرقم 4 ليس IIII.

بدلاً من ذلك، الرقم 4 مكتوب ك IV لأن الرقم 1 قبل 5، نقوم بطرحه ليصبح 4. المبدأ نفسه ينطبق على الرقم 9، والذي مكتوب ك IX هناك ست حالات يتم فيها استخدام الطرح:

- I يمكن أن توضع قبل V و X
- X يمكن أن توضع قبل L و C
- C يمكن أن توضع قبل D و M

اكتب دالة تقبل عددًا صحيحًا وتُعيد رقمه الروماني المكافئ ك سلسلة.

مثال 1

Input: num = 3

Output: "III"

مثال 2

Input: num = 58

Output: "LVIII"

تفسير: $L = 50, V = 5, III = 3$

مثال 3

Input: num = 1994

Output: "MCMXCIV"

تفسير: $M = 1000$, $CM = 900$, $XC = 90$, $IV = 4$

قيود:

```
1 <= num <= 3999
```

```
C:\Users\Samo\OneDrive\Desktop> int_to_roman.py
1 def int_to_roman(num):
2
3     val = [
4         1000, 900, 500, 400,
5         100, 90, 50, 40,
6         10, 9, 5, 4,
7         1
8     ]
9     syb = [
10        "M", "CM", "D", "CD",
11        "C", "XC", "L", "XL",
12        "X", "IX", "V", "IV",
13        "I"
14    ]
15    roman_num = ''
16    i = 0
17    while num > 0:
18        for _ in range(num // val[i]):
19            roman_num += syb[i]
20            num -= val[i]
21        i += 1
22    return roman_num
23
24 # أمثلة على استخدام الدالة
25 print(int_to_roman(3))    # Output: "III"
26 print(int_to_roman(58))  # Output: "LVIII"
27 print(int_to_roman(1994)) # Output: "MCMXCIV"
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS C:\Users\Samo> & C:/Users/Samo/AppData/Local/Programs/Python/Python312/python.exe "C:/Users/Samo/OneDrive/Desktop/???? 2 ??????? ??????.py"
III
LVIII
MCMXCIV
PS C:\Users\Samo>
```

Ln 23, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.4 64-bit