

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:

$$1 \leq num \leq 3999$$

The image shows a Windows 11 desktop with a VS Code editor open. The editor displays a Python script for converting integers to Roman numerals. The script is named 'المسألة 2 المتوسطة.py' and is located at 'C:\Users\Samo\OneDrive\Desktop'. The script defines a function 'int_to_roman(num)' that takes an integer and returns its Roman numeral representation. It uses two lists: 'val' for the values [1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1] and 'syb' for the symbols ['M', 'CM', 'D', 'CD', 'C', 'XC', 'L', 'XL', 'X', 'IX', 'V', 'IV', 'I']. The script includes test cases for 3, 58, and 1994. The terminal shows the output: 'III', 'LVIII', and 'MCMXCIV'. The taskbar at the bottom shows the system clock as 12:23 on 11/11/2023.

```
def int_to_roman(num):  
    val = [  
        1000, 900, 500, 400,  
        100, 90, 50, 40,  
        10, 9, 5, 4,  
        1  
    ]  
    syb = [  
        "M", "CM", "D", "CD",  
        "C", "XC", "L", "XL",  
        "X", "IX", "V", "IV",  
        "I"  
    ]  
    roman_num = ''  
    i = 0  
    while num > 0:  
        for _ in range(num // val[i]):  
            roman_num += syb[i]  
            num -= val[i]  
        i += 1  
    return roman_num  
  
# أمثلة على استخدام الدالة  
print(int_to_roman(3))      # Output: "III"  
print(int_to_roman(58))    # Output: "LVIII"  
print(int_to_roman(1994))  # Output: "MCMXCIV"
```

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>