

Romano para Decimal

Prof. Antonio Carlos Mariani

INE | CTC



UNIVERSIDADE FEDERAL
DE SANTA CATARINA

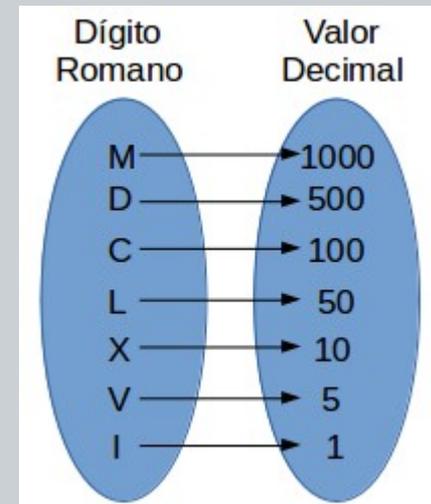
Problema

Converter um número romano para número decimal, a exemplo de:

- CXIII => 113
- DXCIV => 594

O diagrama ao lado apresenta um mapeamento dos dígitos romanos para os correspondentes valores decimais. Este mapeamento corresponde ao conceito de função da matemática, cujo domínio são os dígitos romanos e o codomínio são os número decimais (subconjunto dos naturais).

Há mais de uma forma em Python de construir funções, sendo uma delas com o uso de um dicionário (*dictionary*) como:



```
valores_romanos = {'M':1000, 'D':500, 'C':100, 'L':50, 'X':10, 'V':5, 'I':1}
```

Este dicionário possibilita obter o valor decimal correspondente a um dígito romano por meio de uma expressão simples como:

```
valores_romanos['C']
```

cujo resultado é o valor 100 (o valor associado à letra 'C' no dicionário)

Resolução

Para números romanos triviais, a conversão para decimal pode ser feita simplesmente somando os valores decimais correspondentes aos dígitos romanos, como no exemplo abaixo:

$$\begin{array}{c} \text{C} \quad \text{X} \quad \text{I} \quad \text{I} \quad \text{I} \\ \swarrow \quad \searrow \quad \swarrow \quad \searrow \quad \searrow \\ 100 + 10 + 1 + 1 + 1 = 113 \end{array}$$

Basta, portanto, percorrer os dígitos romanos que formam o número e somar os correspondentes valores decimais, conforme abaixo:

Dígito Romano	Valor Decimal
M	1000
D	500
C	100
L	50
X	10
V	5
I	1

numero_romano.py * x

```
1 valores_romanos = {'M':1000, 'D':500, 'C':100, 'L':50, 'X':10, 'V':5, 'I':1}
2
3 numero_romano = input()
4 numero_decimal = 0
5 for digito_romano in numero_romano:
6     numero_decimal += valores_romanos[digito_romano]
7
8 print(numero_decimal)
```

Resolução

O método anterior falha, contudo, para número menos triviais como DXCIV:

$$\begin{array}{ccccccccc} & & D & X & C & I & V & & \\ & \swarrow & \swarrow & \swarrow & \downarrow & \downarrow & & & \\ 500 & + & 10 & + & 100 & + & 1 & + & 5 = 616 \end{array}$$

pois resulta em 616 quando, em realidade, o valor decimal correspondente é 594. Isso ocorre porque um par como XC equivale a 90 (100-10), e não 110 (10+100) como computado acima. Outros pares similares são:

- IV (4)
- IX (9)
- XL (40)
- XC (90)
- CD (400)
- CM (900)

Dígito Romano	Valor Decimal
M	1000
D	500
C	100
L	50
X	10
V	5
I	1

Há mais de uma forma de tratar estes casos, conforme será visto em seguida.

Resolução

Uma das formas de converter número romanos menos triviais é incluir os pares IV, IX, XL, XC, CD e CM no mapeamento, conforme mostrado ao lado. Em seguida verificamos primeiro os dígitos romanos aos pares e no caso de falhar tratamos individualmente como antes.

1) Começamos com o primeiro par e verificamos que ele não está no mapeamento.

M C D X C I I



decimal = 0

2) Tratamos, então, isoladamente o primeiro dígito romano e avançamos o cursor

M C D X C I I



decimal = 1000

3) Verificamos o próximo par. Ele está no mapeamento de forma que somamos o valor correspondente no número decimal e avançamos o cursor em duas posições.

M C D X C I I



decimal = 1400

4) O próximo par também está no mapeamento. Mesmo comportamento do passo 3)

M C D X C I I



decimal = 1490

Mapeamento

M	→	1000
D	→	500
C	→	100
L	→	50
X	→	10
V	→	5
I	→	1
IV	→	4
IX	→	9
XL	→	40
XC	→	90
CD	→	400
CM	→	900

Resolução

5) O próximo par não está no mapeamento: mesmo comportamento do passo 1) e 2)

M C D X C **I I**

decimal = 1490

6) Tratamos isoladamente o dígito romano e avançamos o cursor.

M C D X C **I** I

decimal = 1491

7) Como não há mais possibilidade de obter um par, trata-se o último dígito romano

M C D X C I **I**

decimal = 1492

O valor decimal

Mapeamento

M	→	1000
D	→	500
C	→	100
L	→	50
X	→	10
V	→	5
I	→	1
IV	→	4
IX	→	9
XL	→	40
XC	→	90
CD	→	400
CM	→	900

Resolução

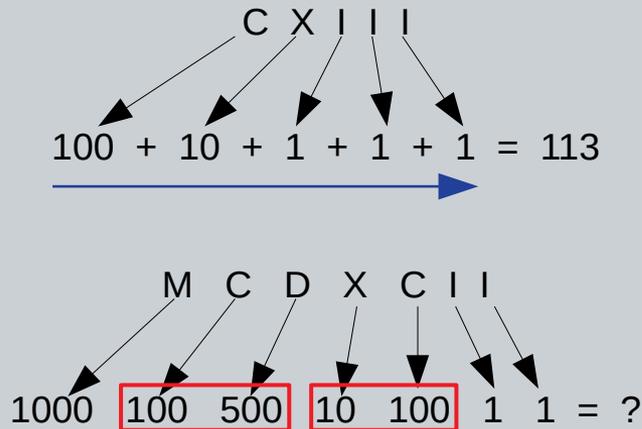
O código Python abaixo implementa o procedimento descrito anteriormente. Nele a variável *i* é utilizada como cursor par indexar os dígitos romanos.

```
numero_romano.py * x
1 valores_romanos = {'M':1000, 'D':500, 'C':100, 'L':50, 'X':10, 'V':5, 'I':1,
2                   'IV':4, 'IX':9, 'XL':40, 'XC':90, 'CD':400, 'CM':900}
3 numero_romano = input()
4 numero_decimal = 0
5 i = 0
6 while i < len(numero_romano):
7     grupo = numero_romano[i:i+2]
8     if grupo in valores_romanos:
9         numero_decimal += valores_romanos[grupo]
10        i += 2
11    else:
12        digito_romano = numero_romano[i]
13        numero_decimal += valores_romanos[digito_romano]
14        i += 1
15 print(numero_decimal)
```

Veja outra estratégia em seguida.

Resolução

Para chegarmos a uma segunda estratégia, veja os dois exemplos abaixo.



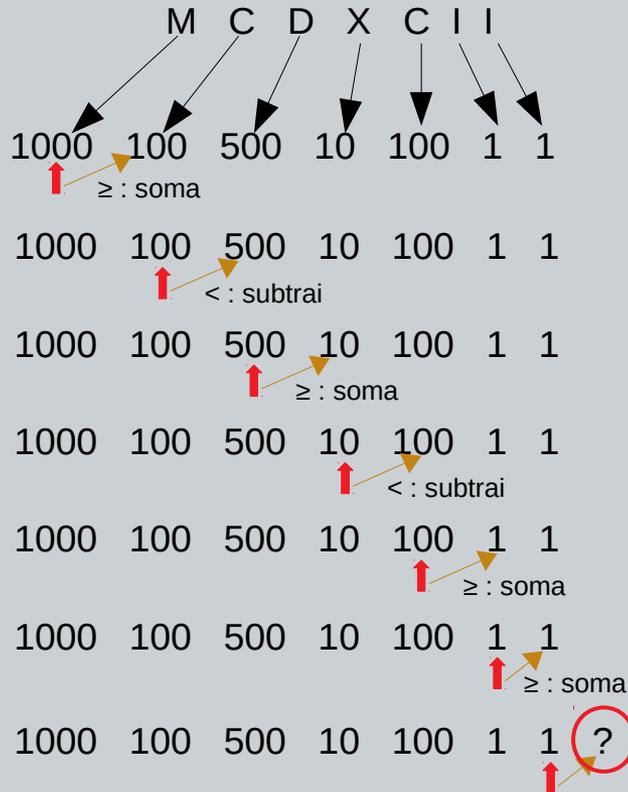
Dígito Romano	Valor Decimal
M	1000
D	500
C	100
L	50
X	10
V	5
I	1

Observe que no primeiro exemplo os valores decimais correspondentes ao dígitos romanos estão em ordem decendente, situação na qual basta somar estes valores para obter o valor decimal. Mas no segundo exemplo há situações na qual esta ordem decrescente não é mantida. Nestes casos o primeiro valor deve ser subtraído, resultando na sequência:

$$\text{decimal} = 1000 - 100 + 500 - 10 + 100 + 1 + 1 = 1492$$

Resolução

Isto sugere um segunda estratégia similar à visto no início de percorrer os dígitos romanos um a um e somar os valores decimais correspondente, mas com uma verificação prévia do valor seguinte: no caso dele ser superior, então o valor atual deve ser subtraído em vez de adicionado.



decimal = 0

decimal = 1000

decimal = 900

decimal = 1400

decimal = 1390

decimal = 1490

decimal = 1491

decimal = 1492

Dígito Romano	Valor Decimal
M	1000
D	500
C	100
L	50
X	10
V	5
I	1

Note que é necessário um tratamento especial para o último dígito romano pois não há um próximo para comparar. Neste caso o valor correspondente pode ser simplesmente somado.

Resolução

O código Python abaixo implementa esta última estratégia.

```
romano.py x
1 valores_romanos = {'M':1000, 'D':500, 'C':100, 'L':50, 'X':10, 'V':5, 'I':1}
2 numero_romano = input()
3
4 decimal = 0
5 for i in range(len(numero_romano)-1): # exclui o último
6     dr = numero_romano[i]
7     dr_seguinte = numero_romano[i+1]
8     if valores_romanos[dr] >= valores_romanos[dr_seguinte]:
9         decimal += valores_romanos[dr]
10    else:
11        decimal -= valores_romanos[dr]
12 decimal += valores_romanos[numero_romano[-1]] #tratamento do último
13
14 print(decimal)
```

Contato

antonio.c.mariani@ufsc.br

CTC/INE



UNIVERSIDADE FEDERAL
DE SANTA CATARINA