Come convertire i numeri arabi in numeri romani con Python

La conversione dei numeri arabi in numeri romani è un esercizio molto comune per chi sta imparando a programmare e desidera familiarizzare con le strutture di controllo e gli algoritmi di base. In questo articolo ti mostrerò come implementare un semplice programma in Python per convertire un numero arabo (decimale) in un numero romano.

Prima di tutto, una breve introduzione ai numeri romani. I simboli principali sono:

Simbolo Valore

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

Le regole principali per la conversione sono:

- Un simbolo più grande che precede uno più piccolo viene sommato (es. VI = 6, XV = 15).
- Un simbolo più piccolo che precede uno più grande viene sottratto (es. IV = 4, IX = 9).

La logica della conversione dei numeri arabi in numeri romani può essere descritta in questi passi:

1. Creiamo una mappa di tutti i valori romani con i loro equivalenti arabi.

2. Iteriamo sulla mappa partendo dal numero arabo più grande e continuiamo a sottrarre dal numero arabo finché possibile, aggiungendo il simbolo romano corrispondente alla stringa di output.

```
# Definiamo una funzione per la conversione
def int_to_roman(num):
    # Creiamo una mappa di conversione numeri romani-
arabi
    roman_map = [
        (1000, 'M'), (900, 'CM'), (500, 'D'), (400,
'CD'),
        (100, 'C'), (90, 'XC'), (50, 'L'), (40,
'XL'),
        (10, 'X'), (9, 'IX'), (5, 'V'), (4, 'IV'),
        (1, 'I')
    ]

    # Stringa vuota che conterrà il risultato
    roman_numeral = ""

    # Iteriamo sulla mappa
    for value, symbol in roman_map:
        # Troviamo quante volte possiamo sottrarre il
valore corrente
        while num >= value:
            roman_numeral += symbol # Aggiungiamo il
simbolo romano corrispondente
            num -= value # Sottraiamo il valore dal
numero arabo

    return roman_numeral

# Test della funzione
```

```
numero_arabo = 1987
numero_romano = int_to_roman(numero_arabo)
print(f"Il numero {numero_arabo} in numeri romani è:
{numero_romano}")
```

Spiegazione del codice:

- 1. Mappa dei Numeri Romani:** La variabile `roman_map` è una lista di tuple, dove ogni tupla rappresenta un numero arabo e il suo simbolo romano corrispondente. L'ordine decrescente è essenziale perché dobbiamo partire dal numero più grande per costruire correttamente il numero romano.
- 2. Iterazione sulla Mappa:** Usiamo un ciclo `for` per scorrere ciascun valore romano nella mappa. Il ciclo `while` interno ci permette di aggiungere il simbolo romano finché il numero arabo è maggiore o uguale al valore associato.
- 3. Costruzione del Risultato:** Ogni volta che sottraiamo un valore dal numero arabo, aggiungiamo il simbolo romano corrispondente alla stringa `roman_number`. Alla fine dell'iterazione, la stringa conterrà il numero romano corretto.

Considerazioni:

- 1. Range dei numeri:** I numeri romani classici vengono utilizzati principalmente per rappresentare numeri fino a poche migliaia. Con il nostro approccio, possiamo gestire numeri fino a circa 3999. Se si volessero numeri più grandi, bisognerebbe considerare l'utilizzo di una notazione diversa (come i numeri romani con barre per indicare moltiplicazioni per 1000).
- 2. Valori negativi e zero:** I numeri romani non hanno una rappresentazione per lo zero o per i numeri negativi. Pertanto, è consigliabile aggiungere un controllo per gestire questi casi.

Conclusione

In questo articolo abbiamo visto come convertire un numero arabo in un numero romano utilizzando Python. Questo tipo di problema è utile per esercitarsi con le strutture di controllo, i cicli e le liste. È anche un ottimo esempio di come affrontare problemi di manipolazione delle stringhe e di costruzione di algoritmi.