

GABRIELE ROMANATO

Menu

Come convertire i numeri arabi in numeri romani con Bash

La conversione dei numeri arabi (i numeri che utilizziamo comunemente, come 1, 2, 3, ecc.) in numeri romani può sembrare un'operazione complessa a prima vista. Tuttavia, con Bash, uno dei linguaggi di scripting più utilizzati in ambienti Linux/Unix, possiamo realizzare un semplice script che automatizza questo processo. In questo articolo, vedremo come fare.

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).

Lo script che scriveremo si baserà su una semplice tabella che associa i numeri arabi ai loro equivalenti romani e procederà sottraendo ripetutamente i valori più grandi possibili fino a completare la conversione.

1. **Tabella di conversione:** Creiamo un array che contiene i numeri arabi e i loro corrispondenti simboli romani.
2. **Ciclo di conversione:** Usando un ciclo `while`, sottraiamo il valore più grande possibile dal numero da convertire e aggiungiamo il corrispondente simbolo romano al risultato.

```
#!/bin/bash

# Funzione per la conversione da numeri arabi a numeri romani
function arabic_to_roman() {
    # Definiamo le corrispondenze tra numeri arabi e romani
    local -a values=(1000 900 500 400 100 90 50 40 10 9 5 4
1)
    local -a numerals=("M" "CM" "D" "CD" "C" "XC" "L" "XL"
"X" "IX" "V" "IV" "I")

    # Otteniamo il numero da convertire
    local num=$1
    local result=""

    # Iteriamo attraverso l'array dei valori e convertiamo
    for i in "${!values[@]}"; do
        while (( num >= values[i] )); do
            result+="${numerals[i]}"
            (( num -= values[i] ))
        done
    done

    echo "$result"
}

# Controlliamo se l'utente ha fornito un numero come
argomento
if [[ $# -eq 0 ]]; then
    echo "Utilizzo: $0 <numero>"
    exit 1
fi

# Verifichiamo se l'argomento è un numero valido
if ! [[ $1 =~ ^[0-9]+$ ]]; then
```

```
    echo "Errore: fornire un numero intero positivo."
    exit 1
fi

# Eseguiamo la funzione con il numero fornito dall'utente
arabic_to_roman "$1"
```

Spiegazione del codice:

1. **Definizione delle tabelle di conversione:** Abbiamo due array, `values` e `numerals`, che contengono rispettivamente i numeri arabi e i corrispondenti simboli romani. Per esempio, 1000 è associato a "M", 900 a "CM" e così via.
2. **Funzione `arabic_to_roman`:** Questa funzione riceve come argomento il numero da convertire. Il ciclo `for` scorre attraverso tutti i valori dell'array e sottrae i valori più grandi dal numero finché è possibile, aggiungendo i corrispondenti simboli romani al risultato.
3. **Validazione dell'input:** Prima di eseguire la conversione, lo script verifica se l'utente ha fornito un argomento e se l'argomento è un numero intero positivo. Se l'input non è valido, lo script stampa un messaggio di errore.

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

La conversione da numeri arabi a numeri romani può essere eseguita facilmente con uno script Bash, utilizzando array per mappare i valori e un ciclo `while` per sottrarre e costruire il numero romano. Questo approccio è semplice e può essere esteso o modificato per adattarsi a esigenze specifiche. Con pochi comandi, possiamo creare uno strumento utile in ambienti Linux per automatizzare la conversione dei numeri.