

Go: verificare se un numero è palindromo

Un numero palindromo è un numero che rimane lo stesso quando le sue cifre sono invertite. Ad esempio, 121, 1331 e 12321 sono tutti numeri palindromi. Se sei un programmatore Go (o Golang), puoi scrivere un programma semplice per verificare se un numero è palindromo. In questo articolo, esploreremo come farlo passo dopo passo.

Prima di iniziare a scrivere codice in Go, è importante capire il concetto di base dietro un numero palindromo. La chiave sta nel capire come ottenere le cifre di un numero e confrontarle in modo appropriato.

1. **Ottenere le cifre:** Dato un numero intero, devi trovare un modo per ottenere le sue cifre. Puoi farlo convertendo il numero in una stringa e quindi esaminando i caratteri della stringa uno alla volta.
2. **Confronto delle cifre:** Una volta ottenute le cifre, devi confrontarle per determinare se il numero è palindromo o meno. Per farlo, puoi confrontare la prima cifra con l'ultima, la seconda con la penultima e così via fino a metà del numero.

Ora che comprendiamo il concetto di base, possiamo procedere con l'implementazione in Go. Ecco un esempio di codice Go per verificare se un numero è palindromo:

```
package main

import (
    "fmt"
    "strconv"
)
```

```

func isPalindrome(num int) bool {
    // Converti il numero in una stringa
    numStr := strconv.Itoa(num)

    // Inizializza gli indici per confrontare le
cifre
    left, right := 0, len(numStr)-1

    for left < right {
        // Confronta le cifre corrispondenti
        if numStr[left] != numStr[right] {
            return false
        }
        // Passa alla coppia successiva di cifre
        left++
        right--
    }

    return true
}

func main() {
    // Testiamo la funzione con alcuni esempi
    fmt.Println(isPalindrome(121))    // True
    fmt.Println(isPalindrome(12321)) // True
    fmt.Println(isPalindrome(12345)) // False
}

```

In questo codice, abbiamo definito una funzione `isPalindrome` che accetta un numero intero come argomento e restituisce `true` se il numero è palindromo e `false` altrimenti.

La funzione prima converte il numero in una stringa utilizzando `strconv.Itoa`. Poi, utilizzando due indici (`left` e `right`), confronta le cifre corrispondenti partendo dai bordi esterni verso il centro. Se in qualsiasi punto le cifre non corrispondono, la funzione restituisce `false`. Se il loop attraversa tutte le cifre senza trovare una discrepanza, il numero è considerato palindromo e la funzione restituirà `true`.

Nel `main()`, abbiamo testato la funzione con alcuni esempi per verificare se funziona correttamente.

Conclusione

Hai ora un codice funzionante per verificare se un numero è palindromo in Go. Questo è un esempio di come puoi utilizzare il linguaggio Go per risolvere un problema matematico relativamente semplice. Puoi ulteriormente espandere questa idea e utilizzare la stessa logica per controllare se una parola è palindroma o esplorare altre varianti di questo problema interessante.