

Lab04 – Iterazione/Stringhe – I parte

Esempio 1 - Indovina l'algoritmo

Qual è l'output di questo programma? Che cosa fa questo algoritmo?

```
package main
import "fmt"
func main() {
    var n int

    fmt.Print("Inserisci un numero: ")
    fmt.Scan(&n)

    for i := 0; i <= n; i += 2 {
        fmt.Print(i, " ")
    }

    fmt.Println()
}
```

Esempio 2 - Indovina l'algoritmo

Qual è l'output di questo programma? Che cosa fa questo algoritmo? Quali sono le differenze con il programma precedente?

```
package main

import "fmt"

func main() {
    var n int

    fmt.Print("Inserisci un numero: ")
    fmt.Scan(&n)

    for i := 1; i <= n; i *= 2 {
        fmt.Print(i, " ")
    }
    fmt.Println()
}
```

Esempio 3 - Trova gli errori

Questo programma è errato e non produce l'output descritto nel commento la package, correggilo (3 errori: 1 di sintassi, 2 di logica del programma).

SUGGERIMENTO: Puoi aiutarti copiando il codice in un file go ed eseguendolo

```
/*
Il package stampa una sequenza di numeri

Dato un numero n inserito da tastiera, il package stampa tutti i numeri compresi tra 1 e n
(estremi inclusi).
La sequenza è prodotta su un'unica linea di output, con i numeri separati da uno spazio.
*/
package main

import "fmt"

func main() {
    var n int

    fmt.Print("Inserisci un numero: ")
    fmt.Scan(&n)

    for i := 1; i < n {
        fmt.Print(i)
        i++
    }

    fmt.Println()
}
```

Esercizio 4 – Quadrati di Asterischi

Implementate un programma che legge un intero n inserito da tastiera e quindi disegna un quadrato di n asterischi intervallati da spazi.

SUGGERIMENTO

Utilizzate due cicli `for` nidificati.

Esempi di funzionamento (in grassetto il valore inserito dall'utente):

```
>go run quadratiAsterischiSpaziati.go
Inserisci un numero: 4
* * * *
```

```
* * * *
* * * *
* * * *
```

```
>go run quadratiAsterischiSpaziati.go
```

```
Inserisci un numero: 3
```

```
* * *
* * *
* * *
```

Esercizio 5 – Quadrati Alternati

Scrivete un programma che, letto un numero intero n , produca in output una “figura” di n righe costituite ciascuna da n simboli intervallati da spazi, alternando fra loro righe costituite solo da simboli $*$ (asterisco) intervallati da spazi e righe costituite solo da simboli $+$ (più) intervallati da spazi.

SUGGERIMENTO

Utilizzate due cicli `for` nidificati e stampate i simboli $+$ e $*$ (intervallati da spazi) nella posizione corretta. Verificate il fatto che la posizione sia corretta o meno in base al valore della variabile di indice del ciclo più esterno.

Assumendo che:

- le righe della figura siano numerate a partire da 0 (dall’alto verso il basso);
- le righe della figura siano identificate/associate dai/ai valori della variabile di indice del ciclo più esterno;

nel ciclo di seguito riportato il blocco di codice (1) viene eseguito per le righe pari, mentre il blocco di codice (2) viene eseguito per le righe dispari.

```
for i:=0; i<n; i++ { // ciclo più esterno
    if i%2 == 0 {
        // blocco di codice (1)
    } else {
        // blocco di codice (2)
    }
}
```

Esempi di funzionamento (in grassetto il valore inserito dall’utente):

```
>go run quadratiAlternati.go
```

```
Inserisci un numero: 5
```

```

* * * * *
+ + + + +
* * * * *
+ + + + +
* * * * *

>go run quadratiAlternati.go
Inserisci un numero: 10
* * * * *
+ + + + +
* * * * *
+ + + + +
* * * * *
+ + + + +
* * * * *
+ + + + +
* * * * *
+ + + + +

```

Esercizio 6 – Variazione Quadrati Alternati

Scrivete un programma che, letto da standard input un numero intero n , produca in output una “figura” di n righe costituite ciascuna da n simboli intervallati da spazi, alternando fra loro: (1) righe costituite solo da simboli * (asterisco) intervallati da spazi, (2) righe costituite solo da simboli + (più) intervallati da spazi, e (3) righe costituite solo da simboli o (lettera o) intervallati da spazi

Esempi di funzionamento (in grassetto il valore inserito dall’utente):

```

>go run quadratiAlternati.go
Inserisci un numero: 5
* * * * *
+ + + + +
o o o o o
* * * * *
+ + + + +

>go run quadratiAlternati.go
Inserisci un numero: 10
* * * * *

```

```

+ + + + + + + + + +
o o o o o o o o o o
* * * * * * * * * *
+ + + + + + + + + +
o o o o o o o o o o
* * * * * * * * * *
+ + + + + + + + + +
o o o o o o o o o o
* * * * * * * * * *

```

Esercizio 7 – Variazione Quadrati di Asterischi

Provate ora a disegnare il quadrato di asterischi (vedi Esecizio 5) con l'interno riempito di + (i simboli sono anche in questo caso separati da spazi). Ovvero il programma dovrebbe funzionare come segue:

SUGGERIMENTO

Utilizzate due cicli `for` nidificati e stampate i simboli + e * (intervallati da spazi) nella posizione corretta. Verificate il fatto che la posizione sia corretta o meno in base al valore delle variabili di indice. Questa volta, per determinare se la posizione è corretta o meno, devono essere considerati i valori di entrambe le variabili di indice.

Esempi di funzionamento (in grassetto il valore inserito dall'utente):

```
>go run quadratiAsterischiPiu.go
```

```
Inserisci un numero: 4
```

```

* * * *
* + + *
* + + *
* * * *

```

```
>go run quadratiAsterischiPiu.go
```

```
Inserisci un numero: 3
```

```

* * *
* + *
* * *

```

```
>go run quadratiAsterischiPiu.go
```

```
Inserisci un numero: 7
```

```

* * * * * * *
* + + + + + *

```

```

* + + + + + *
* + + + + + *
* + + + + + *
* + + + + + *
* + + + + + *
* * * * * *

```

Esercizio 8 - Figure

Scrivete altri tre programmi simili a quelli degli esercizi precedenti.

Tutti e 3 i programmi prendono in input un intero n e producono in output una figura di n righe e n colonne, costituita da simboli $*$ (asterisco) e $+$ (più) non intervallati da spazi. La differenza fra un programma e l'altro è il modo in cui sono alternati i simboli, come indicato negli esempi che seguono.

Primo programma:

Ognuna delle n righe è composta da simboli $*$ e $+$ alternati.

Esempi di funzionamento (in grassetto il valore inserito dall'utente):

```

> go run figure1.go
Inserisci un numero: 5
*+*+*
*+*+*
*+*+*
*+*+*
*+*+*

> go run figure1.go
Inserisci un numero: 7
*+*+*+*
*+*+*+*
*+*+*+*
*+*+*+*
*+*+*+*
*+*+*+*
*+*+*+*

```

Secondo programma:

Ognuna delle n righe è composta da coppie di simboli $*$ e $+$ (ovvero da $**$ e $++$) alternati, fino a raggiungere gli n simboli. Attenzione che, se n è dispari, la riga sarà terminata da un solo simbolo $*$ (asterisco) o $+$ (più).

Esempi di funzionamento (in grassetto il valore inserito dall'utente):

```
>go run figure2.go
Inserisci un numero: 6
**++**
++**++
**++**
++**++
**++**
++**++
>go run figure2.go
Inserisci un numero: 7
**++**+
++**++*
**++**+
++**++*
**++**+
++**++*
**++**+
```

Terzo programma:

Il quadrato di lato n è composto da due triangoli rettangoli isosceli i cui lati sono lunghi $n-1$. In sostanza in questa figura, i simboli sopra la diagonale del quadrato sono $+$, quelli sotto sono $*$, mentre sulla diagonale ci sono o .

Esempi di funzionamento (in grassetto il valore inserito dall'utente):

```
>go run figure3.go
Inserisci un numero: 7
o+++++
*o++++
**o+++
***o++
```

```
****O++
*****O+
*****O
>go run figure3.go
Inserisci un numero: 9
O+++++++
*O+++++++
**O+++++++
***O+++++++
****O+++++
*****O+++
*****O++
*****O+
*****O
```