

Laboratorio 3 - Cicli annidati

1 Qual è l'output?

Quanti asterischi stampa il seguente programma?

```
package main

import "fmt"

func main() {
    var n int = 3

    var i, j int
    for i = 0; i < n; i++ {
        for j = 0; j < n; j++ {
            fmt.Print("*")
        }
    }
    fmt.Println()
}
```

2 Qual è l'output?

Quanti asterischi stampa il seguente programma?

```
package main

import "fmt"

func main() {
    var n int = 3

    var i, j, z int
    for i = 0; i < n; i++ {
        for j = 0; j < n; j++ {
            for z = 0; z < n; z++ {
                fmt.Print("*")
            }
        }
    }
    fmt.Println()
}
```

3 Qual è l'output?

Quanti asterischi stampa il seguente programma?

```
package main

import "fmt"

func main() {
    var n int = 3

    var i, j int
    for i = 0; i < n; i++ {
        for j = i+1; j < n; j++ {
            fmt.Print("*")
        }
    }
}
```

```
    }  
    }  
    fmt.Println()  
}
```

4 Trova l'errore

Questo programma dovrebbe stampare 4 asterischi ma non funziona correttamente. Qual è l'errore?

```
package main  
  
import "fmt"  
  
func main() {  
    var n int = 2  
  
    var i, j int  
    for i = 0; i < n; i++ {  
        for j = 0; j < n; i++ {  
            fmt.Print("**")  
        }  
    }  
    fmt.Println()  
}
```

5 Trova l'errore

Questo programma dovrebbe stampare 4 asterischi ma non funziona correttamente. Qual è l'errore?

```
package main  
  
import "fmt"  
  
func main() {  
    var n int = 2  
  
    var i, j int  
    for i = 0; i < n; i++ {  
        for j = 0; j < n; j-- {  
            fmt.Print("**")  
        }  
    }  
    fmt.Println()  
}
```

6 Quadrato di asterischi

Scrivere un programma che legga da **standard input** un numero intero `n` e stampi a video un quadrato di `n` asterischi intervallati da spazi come mostrato nell'**Esempio di esecuzione**.

Suggerimento: potete utilizzare due cicli `for` annidati.

Esempio d'esecuzione:

```
$ go run quadratoasterischi.go  
Inserisci un numero: 3  
* * *  
* * *  
* * *
```

7 Quadrato a righe alterne (1)

Scrivere un programma che legga da **standard input** un numero intero `n` e che, come mostrato nell'**Esempio di esecuzione**, stampi a video un quadrato 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: potete utilizzare due cicli `for` annidati ed utilizzare l'operatore `%` per distinguere le righe pari da quelle dispari.

Esempio d'esecuzione:

```
$ go run quadrato_righe_alterne_1.go
Inserisci un numero: 5
* * * * *
+ + + + +
* * * * *
+ + + + +
* * * * *
```

8 Quadrato a righe alterne (2)

Scrivere un programma che legga da **standard input** un numero intero `n` e che, come mostrato nell'**Esempio di esecuzione**, stampi a video un quadrato di `n` righe costituite ciascuna da `n` simboli intervallati da spazi, alternando fra loro righe costituite solo da simboli `*` (asterisco) intervallati da spazi, righe costituite solo da simboli `+` (più) intervallati da spazi e righe costituite solo da simboli `o` (lettera o) intervallati da spazi.

Esempio d'esecuzione:

```
$ go run quadrato_righe_alterne_2.go
Inserisci un numero: 5
* * * * *
+ + + + +
o o o o o
* * * * *
+ + + + +
```

9 Quadrato: perimetro ed area interna

Scrivere un programma che legga da **standard input** un numero intero `n` e che, come mostrato nell'**Esempio di esecuzione**, stampi a video un quadrato di lato `n` in cui:

- il perimetro è rappresentato con il carattere `*` (asterisco);
- l'area interna è rappresentata con il carattere `+` (più);
- i caratteri di ogni riga del quadrato sono separati tra di loro da un carattere spazio.

Esempio d'esecuzione:

```
$ go run quadrato_perimetro_area.go
Inserisci un numero: 4
* * * *
* + + *
* + + *
* * * *
```

10 Quadrato a colonne alterne (1)

Scrivere un programma che legga da **standard input** un numero intero `n` e che, come mostrato nell'**Esempio di esecuzione**, stampi a video un quadrato di `n` righe costituite ciascuna da `n` simboli intervallati da spazi, alternando fra loro colonne costituite da simboli `*` (asterisco) a colonne costituite da simboli `+` (più).

Esempio d'esecuzione:

```
$ go run quadrato_colonne_alterne_1.go
Inserisci un numero: 5
* + * + *
* + * + *
* + * + *
* + * + *
* + * + *
```

11 Quadrato a colonne alterne (2)

Scrivere un programma che legga da **standard input** un numero intero `n` e che, come mostrato nell'**Esempio di esecuzione**, stampi a video un quadrato di `n` righe costituite ciascuna da `n` simboli intervallati da spazi, alternando fra loro 2 colonne costituite da simboli `*` (asterisco) a 2 colonne costituite da simboli `+` (più). In particolare, se `n` è dispari solo una delle due colonne più a destra sarà stampata.

Esempio d'esecuzione:

```
$ go run quadrato_colonne_alterne_2.go
Inserisci un numero: 7
* * + + * * +
* * + + * * +
* * + + * * +
* * + + * * +
* * + + * * +
* * + + * * +
* * + + * * +

$ go run quadrato_colonne_alterne_2.go
Inserisci un numero: 5
* * + + *
* * + + *
* * + + *
* * + + *
* * + + *
```

12 Quadrato con diagonale

Scrivere un programma che legga da **standard input** un numero intero `n` e che, come mostrato nell'**Esempio di esecuzione**, stampi a video un quadrato di lato `n` in cui:

- una diagonale è rappresentata utilizzando il carattere `o` (lettera o);
- l'area del quadrato al di sotto della diagonale è rappresentata con il carattere `*` (asterisco);
- l'area del quadrato al di sopra della diagonale è rappresentata con il carattere `+` (più);
- i caratteri di ogni riga del quadrato sono separati tra di loro da un carattere spazio.

Esempio d'esecuzione:

```
$ go run quadrato_con_diagonale.go
Inserisci un numero: 7
o + + + + +
* o + + + +
* * o + + +
* * * o + +
* * * * o +
* * * * * o +
```

13 Triangoli

Scrivere un programma che legga da **standard input** un intero `n > 1` e stampi, utilizzando il carattere `*`, il perimetro di due triangoli rettangoli con base e altezza di lunghezza `n`, posizionati come mostrato nell'**Esempio d'esecuzione**.

Esempio d'esecuzione:

```
$ go run triangoli.go
2
**
 *
 *
**

$ go run triangoli.go
3
***
**
 *
 *
**
***

$ go run triangoli.go
4
****
* *
**
 *
 *
**
* *
****

$ go run triangoli.go 6
*****
*   *
*   *
*   *
**
 *
 *
**
* *
* *
*   *
*****
```

14 Albero

Scrivere un programma che legga da **standard input** un intero `n` e, come mostrato nell'**Esempio d'esecuzione**, stampi a video un albero utilizzando il carattere `*` (asterisco) per rappresentarne la chioma ed il carattere `+` (più) per rappresentarne il tronco:

- La chioma dell'albero deve essere alta `n` righe e, nel punto di larghezza massima, larga `2*n-1` colonne.
- Il tronco dell'albero deve essere rappresentato con un quadrato di altezza e larghezza pari a `3` caratteri.
- Il tronco dell'albero deve essere centrato rispetto alla chioma dell'albero.

Se `n<=0`, il programma stampa solo il tronco dell'albero.

Esempio d'esecuzione:

```
$ go run albero.go
```

```
7
```

```
      *
    ***
  *****
*****
*****
*****
*****
*****
+++
+++
+++
```

```
$ go run albero.go
```

```
4
```

```
      *
    ***
  *****
*****
+++
+++
+++
```

```
$ go run albero.go
```

```
1
```

```
      *
+++
+++
+++
```

```
$ go run albero.go
```

```
0
```

```
+++
+++
+++
```

```
$ go run albero.go
```

```
-1
```

```
+++
+++
+++
```