

## Laboratorio 2 - Variabili e IO

# 1 Scan

Il programma riportato di seguito legge da **standard input** 4 numeri interi.

- ▶ Cosa succede se inserite l'input su righe diverse?
- ▶ Cosa succede se inserite meno numeri di quelli richiesti?
- ▶ Cosa succede se inserite più numeri di quelli richiesti?
- ▶ Cosa succede se inserite un valore diverso da un numero intero (numero reale, lettera, parola, ...)?

# 1 Scan

```
package main

import "fmt"

func main() {

    var n1, n2, n3, n4 int

    fmt.Scan(&n1, &n2, &n3, &n4)

    fmt.Println(n1, n2, n3, n4)
}
```

## 2 Print o Println

Di seguito sono riportati due programmi: in entrambi sono dichiarate 2 variabili a e b di cui vengono stampati i valori. Notate differenze nell'output prodotto? Cosa cambia tra `Print` e `Println`?

## 2 Print o Println (programma 1)

```
package main

import "fmt"

func main() {
    var a int = 5
    var b float64 = 3.14

    fmt.Print("Valore di a:", a, "capito? Te lo dico due volte:", a, a, "...\\n")
    fmt.Print("Valore di b:", b, b, "\\n")
}
```

## 2 Print o Println (programma 2)

```
package main

import "fmt"

func main() {
    var a int = 5
    var b float64 = 3.14

    fmt.Println("Valore di a:", a, "capito? Te lo dico due volte:", a, a, "...")
    fmt.Println("Valore di b:", b, b)
}
```

### 3 Qual è l'output?

Supponendo che l'input fornito da **standard input** sia

5 6

cosa dovrebbe produrre in output il seguente programma?

### 3 Qual è l'output?

```
package main

import "fmt"

func main() {

    var a, b int

    fmt.Scan(&a, &b)

    var r int

    r = a - b

    fmt.Println(r)

}
```



## 4 Qual è l'output?

Cosa dovrebbe stampare il seguente programma?

```
package main

import "fmt"

func main() {
    var a int = 10
    var b int = 20
    a = a + b
    var c int = a + b
    fmt.Println(c)
}
```

## 5 Trova l'errore

Questo programma dovrebbe stampare la somma di tre numeri interi  $a$ ,  $b$  e  $c$ , ma contiene degli errori. Correggere gli errori e verificare che l'esecuzione produca l'output desiderato.

## 5 Trova l'errore

```
package main

import "fmt"

func main() {
    var a
    a = 10

    var b, d int
    b = 20

    c = 30
    var c int

    var d int = a + b + c

    fmt.Println((d)
}
```

## 6 Area e perimetro rettangolo

Scrivere un programma che legga da **standard input** le misure dell'altezza e della base di un rettangolo e ne calcoli il perimetro e l'area.

Esempio d'esecuzione:

```
$ go run rettangolo.go
Inserisci la base: 20
Inserisci l'altezza: 10
Perimetro = 60
Area = 200
```

## 7 Area cerchio

Scrivere un programma che legga da **standard input** il raggio di un cerchio e ne calcoli circonferenza e area.

*Suggerimento:* l'area del cerchio si calcola facendo  $\text{raggio} \times \text{raggio} \times \text{pi\_greco}$ , mentre la circonferenza facendo  $2 \times \text{raggio} \times \text{pi\_greco}$ . Il valore numerico di `pi_greco` è memorizzato nella costante `Pi` del package `math`, a cui ci si può riferire con `math.Pi`.

### Esempio d'esecuzione:

```
$ go run cerchio.go  
Raggio = 2.5  
Circonferenza = 15.707963267948966  
Area = 19.634954084936208
```

## 8 Convertitore Km - miglia

Scrivere un programma che legga da **standard input** una distanza in Km ed effettui la conversione di tale distanza in miglia ( $1 \text{ Km} = 0.62137 \text{ mi}$ ).

Esempio d'esecuzione:

```
$ go run convertitore.go
```

```
Distanza (Km) = 12
```

```
Distanza (mi) = 7.45644
```

## 9 Età

Scrivere un programma che legga da **standard input** le età di due persone (esprese in anni) e calcoli:

- ▶ la somma delle età inserite;
- ▶ la media delle età inserite;
- ▶ la media delle età inserite arrotondata per difetto all'intero inferiore;
- ▶ la media delle età inserite arrotondata per eccesso all'intero superiore;
- ▶ la somma e la media delle età che le due persone avranno fra 10 anni.

*Suggerimento:* la media arrotondata per difetto può essere calcolata usando la funzione `math.Floor` del package `math` nel seguente modo:

```
var mediaArrotondataDifetto float64 = math.Floor(media)
```

Similarmente, la media arrotondata per eccesso può essere calcolata usando la funzione `math.Ceil` nel seguente modo:

```
var mediaArrotondataEccesso float64 = math.Ceil(media)
```

## 9 Età (esempio esecuzione)

### Esempio d'esecuzione:

```
$ go run calcoloeta.go
```

```
Età persona 1 = 15
```

```
Età persona 2 = 20
```

```
Somma età = 35
```

```
Media età = 17.5
```

```
Media età arrotondata per difetto all'intero inferiore = 17
```

```
Media età arrotondata per eccesso all'intero superiore = 18
```

```
Somma età a 10 anni = 55
```

```
Media età a 10 anni = 27.5
```



## 10 Area poligono regolare

Scrivere un programma che legga da **standard input** due numeri interi che chiameremo  $n$  e  $l$  e calcoli l'area di un poligono regolare con  $n$  lati di lunghezza  $l$ .

*Suggerimento:* l'area di un poligono regolare può essere calcolata utilizzando le funzioni `math.Pow` (per il calcolo della potenza) e `math.Tan` (per il calcolo della tangente di un angolo) del package `math` nel seguente modo:

```
var area float64 = (n * math.Pow(l, 2)) / (4 * math.Tan(math.Pi/n))
```

### Esempio d'esecuzione:

```
$ go run areapoligono.go
```

```
Inserisci il numero di lati del poligono: 6
```

```
Inserisci la lunghezza dei lati del poligono: 3
```

```
Area calcolata: 23.382685902179844
```

```
$ go run areapoligono.go
```

```
Inserisci il numero di lati del poligono: 4
```

```
Inserisci la lunghezza dei lati del poligono: 3
```

```
Area calcolata: 9
```

## 11 Da secondi a ore, minuti, secondi

Scrivere un programma che prenda in ingresso un valore intero rappresentante una quantità di tempo in secondi e restituisca il numero di ore, minuti e secondi corrispondenti

Esempio di esecuzione:

```
$ go run convertisec.go  
Secondi? 15634  
h:m:sec - 4:20:34
```

```
$ go run convertisec.go  
Secondi? 134  
h:m:sec - 0:2:14
```