

Laboratorio 3 - Selezione

1 Qual è l'output?

Qual è l'output del seguente programma?

```
package main
```

```
import "fmt"
```

```
func main() {  
    var (  
        a, b int = 10, 20  
        c int = 30  
    )  
    if a > b {  
        a = b  
    } else {  
        b = a  
    }  
    c = c + b + a  
    fmt.Println(a, b, c)  
}
```

2 Qual è l'output?

Supponendo che l'utente inserisca da **standard input**

5 8 6

qual è l'output del seguente programma? Che cosa calcola?

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var a, b, c int
```

```
    var m int
```

```
    fmt.Scan(&a, &b, &c)
```

```
    if a < b {
```

```
        if a < c {
```

```
            m = a
```

```
        } else {
```

3 Qual è l'output?

Confrontate i programmi che seguono. Cosa producono in output? Il loro funzionamento è identico?

```
package main

import "fmt"

func main() {

    var a, b int = 10, 10

    if a <= b {

        fmt.Println("a <= b")

    } else {

        fmt.Println("a > b")
    }
}
```

4 Qual è l'output?

Confronta i programmi che seguono. Cosa stampano nel caso in cui l'utente inserisca da **standard input** il valore 150? E se invece inserisse 40? I due programmi funzionano allo stesso modo?

```
package main

import "fmt"

func main() {

    var a int

    fmt.Scan(&a)

    if a < 100 {
        fmt.Println("a minore di 100")
    } else if a < 200 {
        fmt.Println("a compreso tra 100 e 200")
    } else {
        fmt.Println("a maggiore o uguale a 200")
    }
}
```

5 Intero con segno

Scrivere un programma che legge da **standard input** un numero intero n (specificato senza segno se maggiore o uguale a 0) e stampi a video il numero con segno.

Esempio d'esecuzione:

```
$ go run interoconsegno.go  
Inserisci numero: 5  
+5
```

```
$ go run interoconsegno.go  
Inserisci numero: 0  
0
```

```
$ go run interoconsegno.go  
Inserisci numero: -5  
-5
```

6 Multiplo di 10

Scrivere un programma che legge da **standard input** un numero intero `n` e verifica se il numero è multiplo di 10.

Suggerimento: per verificare se un numero sia multiplo di 10 potete utilizzare l'operatore `%` che calcola il resto della divisione tra interi.

Esempio d'esecuzione:

```
$ go run multiplo10.go
Inserisci numero: 15
15 non è multiplo di 10
```

```
$ go run multiplo10.go
Inserisci numero: 20
20 è multiplo di 10
```

7 Intervallo

Scrivere un programma che legga da **standard input** un voto v da 0 a 100 e stampi:

* Insufficiente se il voto è inferiore a 60 ($v < 60$) * Sufficiente se il voto è compreso tra 60 e 70 ($v \geq 60$ e $v < 70$) * Buono se il voto è compreso tra 70 e 80 ($v \geq 70$ e $v < 80$) * Distinto se il voto è compreso tra 80 e 90 ($v \geq 80$ e $v < 90$) * Ottimo se il voto è compreso tra 90 e 100 ($v \geq 90$ e $v \leq 100$) * Errore se il voto è negativo o superiore a 100

Esempio d'esecuzione:

```
$ go run voto.go
```

```
Inserisci il voto: 75
```

```
Buono
```

```
$ go run voto.go
```

```
Inserisci il voto: 90
```

```
Ottimo
```

```
$ go run voto.go
```

```
Inserisci il voto: 110
```

```
Errore
```


8 Fizz Buzz

Scrivere un programma che legge da **standard input** un numero intero e stampa "Fizz" se il numero è multiplo di 3, "Buzz" se il numero è multiplo di 5, "Fizz Buzz" se è multiplo sia di 3 sia di 5, niente altrimenti.

Esempio d'esecuzione:

```
$ go run fizzbuzz.go  
Inserisci un numero: 5  
Buzz
```

```
$ go run fizzbuzz.go  
Inserisci un numero: 4
```

```
$ go run fizzbuzz.go  
Inserisci un numero: 15  
Fizz Buzz
```

```
$ go run fizzbuzz.go  
Inserisci un numero: 6  
Fizz
```

9 Pari o dispari

Scrivere un programma che legge da **standard input** un intero `n` e stampa a video se il numero è pari o dispari.

Esempio d'esecuzione:

```
$ go run paridispari.go
Inserisci un numero: 10
10 è pari
```

```
$ go run paridispari.go
Inserisci un numero: 11
11 è dispari
```

10 Divisione

Scrivere un programma che legga da **standard input** due numeri interi a e b e calcoli il risultato della divisione a/b . Se b è uguale a 0, il programma stampa Impossibile.

Esempio d'esecuzione:

```
$ go run divisione.go
Inserisci due numeri:
5 2
Quoziente = 2.5
```

```
$ go run divisione.go
Inserisci due numeri:
5 0
Impossibile
```

11 Angoli di un triangolo

Scrivere un programma che legga da **standard input** le ampiezze di due angoli di un triangolo e stampi, se possibile, l'ampiezza del terzo angolo.

Suggerimento: ricordatevi che in un triangolo la somma delle ampiezze degli angoli interni è sempre 180° .

Esempio d'esecuzione:

```
$ go angolitriangolo.go
```

```
Inserire le ampiezze dei due angoli: 50 60
```

```
Ampiezza terzo angolo = 70°
```

```
$ go angolitriangolo.go
```

```
Inserire le ampiezze dei due angoli: 150 70
```

```
I due angoli non appartengono ad un triangolo
```

12 Conversioni

Scrivere un unico programma che: - legga da **standard input** un valore intero che specifica il tipo di conversione da effettuare:

- 1: secondi (inseriti dall'utente) in ore
- 2: secondi inseriti dall'utente in minuti
- 3: minuti inseriti dall'utente in ore
- 4: minuti inseriti dall'utente in secondi
- 5: ore inserite dall'utente in secondi
- 6: ore inserite dall'utente in minuti
- 7: minuti inseriti dall'utente in giorni e ore
- 8: minuti inseriti dall'utente in anni e giorni

gestendo l'inserimento di un valore di scelta non compreso tra 1 e 8;

- ▶ legga da **standard input** un valore reale da convertire;
- ▶ stampi a video il valore convertito.

Esempio d'esecuzione:

```
$ go run conversioni.go
```

Scegli la conversione:

1) secondi -> ore

13 Retta

Scrivere un programma che legga da **standard input** 4 valori a virgola mobile:

- i primi due valori sono il coefficiente angolare m e il termine noto q di una retta r : $y = m*x + q$
- il terzo e il quarto valore sono le coordinate px e py di un punto $P(px,py)$

Il programma deve determinare se il punto P sta sopra o sotto la retta od appartiene ad essa, e stampare a video il relativo messaggio.

Suggerimento: un punto appartiene ad una retta se sostituendo le sue coordinate nell'equazione della retta l'uguaglianza è verificata. Un punto sta sopra una retta se sostituendo il valore dell'ascissa nell'equazione della retta si ottiene $y < py$.

```
$ go run retta.go
Inserisci m e q: 1 0
Inserisci x e y: 5 5
Il punto appartiene alla retta
```

```
$ go run retta.go
Inserisci m e q: 1 1
Inserisci x e y: 5 5
Il punto sta sotto la retta
```