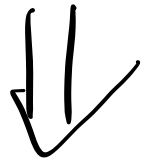
ESERCIZIO

- Scrivere una funzione che date due slice di interi x e y restituisce una nuova slice che contiene solo gli elementi di x che compaiono in y.

x | 3 | 7 | 2 | 3 | 1 |

y | 3 | 7 | 3 | 3 | 3 | 2 |



| 3 | 2 | 3 |

SOLUZIONE 1

```
func f(x,y []int) []int {
    var z []int
    for _, ex := range x {
        var i int
        for i=0; i<len(y); i++ {
            if y[i]==ex {
                break
            }
        }
        if i<len(y) {
            z=append(z,ex)
        }
    }
    return z
}
```

SOLUZIONE 2

```
func contains(x []int, v int) bool {
    for _, ex := range x {
        if ex == v {
            return true
        }
    }
    return false
}

func f(x, y []int) []int {
    var z []int
    for _, ex := range x {
        if contains(y, ex) {
            z = append(z, ex)
        }
    }
    return z
}
```

ARGOMENTI SULLA RIGA DI COMANDO

```
$> go build pippo.go
$> ./pippo a b 15 26
```

Scriviamo un programma somma2.go

```
$> ./somma2 12a 15 120
147
$>
```

LIBRERIA os

os.Args []string

- contiene come primo elemento il nome del programma eseguibile

e che successivi
elementi gli (eventuali)
argomenti sulla riga
di comando.

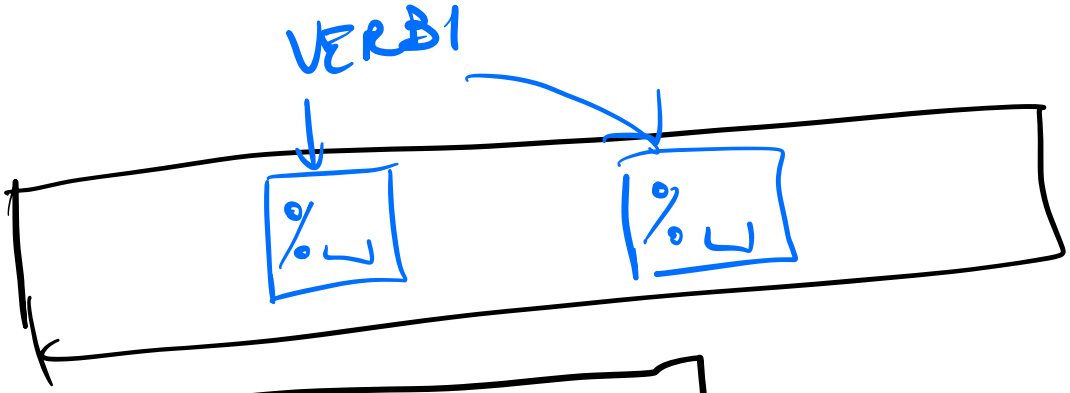
```
func main() {
    somma := 0
    for i := 1; i < len(os.Args); i++ {
        x, err := strconv.Atoi(os.Args[i])
        if err != nil {
            fmt.Println("L'argomento",
            os.Args[i], "non è un
            intero e sarà ignorato")
        } else {
            somma += x
        }
    }
    fmt.Println(somma)
```

}

Printf & friends

fmt.Printf([ ] , arg1, arg2, ...)

stringa di formato



VERBI	POSSIBILI
%d	intero
%c	carattere
%f	float
%f	float
%g	
%s	string

%v

default for ast

```
fmt.Println("L'argomento",
os.Args[i], "non è un
intero e sarà ignorato")
```

```
fmt.Printf("L'argomento %s non è intero
e sarà ignorato\n", os.Args[i])
```

```
fmt.Printf("%d + %d = %d\n",
x, y, x+y)
```

%-15s

↓

LARGHEZZA

x := "Pippo"

fmt.Printf("Ciao %8s\n", x)

%f

%.2f ⟵ solo 2 cifre decimali

%8.2f ⟵ 2 cifre decim. e 8 caratteri in tutto

fmt.Sprintf