

var x, y [10] int

len(x)

x[0], ..., x[9]

y = x

var x [] int

x = [] int {5, 0, 9, 6}

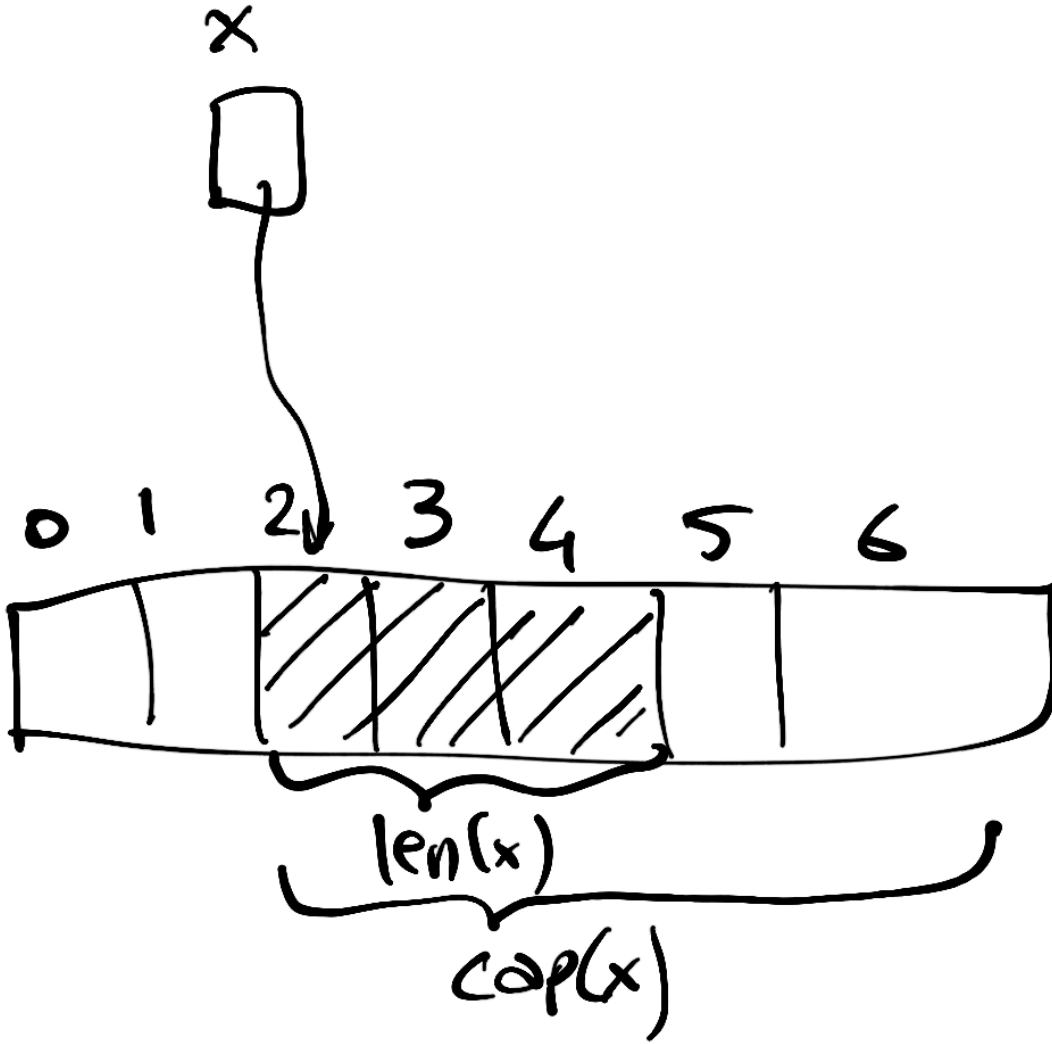
x = make([] int, 10)

len(x)

x[0], ..., x[len(x)-1]

x = append(x, 7)

x []int



[]int

```
struct {  
    len int  
    cap int  
    data *int  
}
```

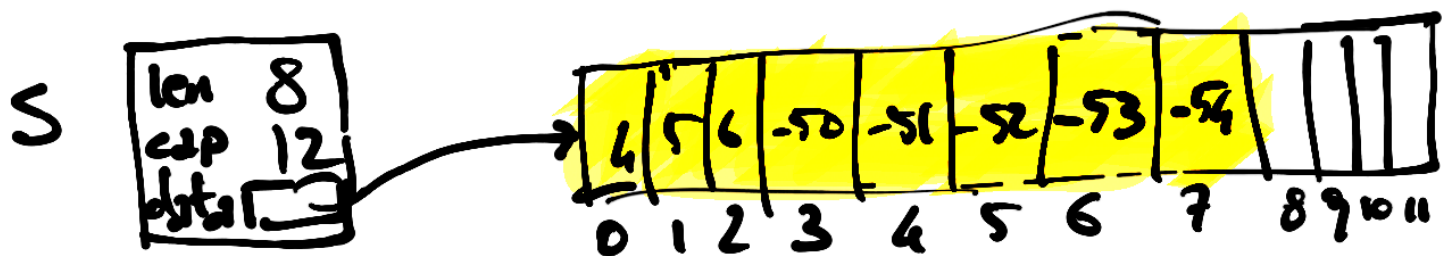
SUBSLICING DI ARRAY

UN MODO PER CREARE UNA SLICE

var a [10]int = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

var s []int
s = a [3:6]

s = append(s, -50)
s = append(s, -51)
s = append(s, -52)
s = append(s, -53)
s = append(s, -54)



var s, t [] int

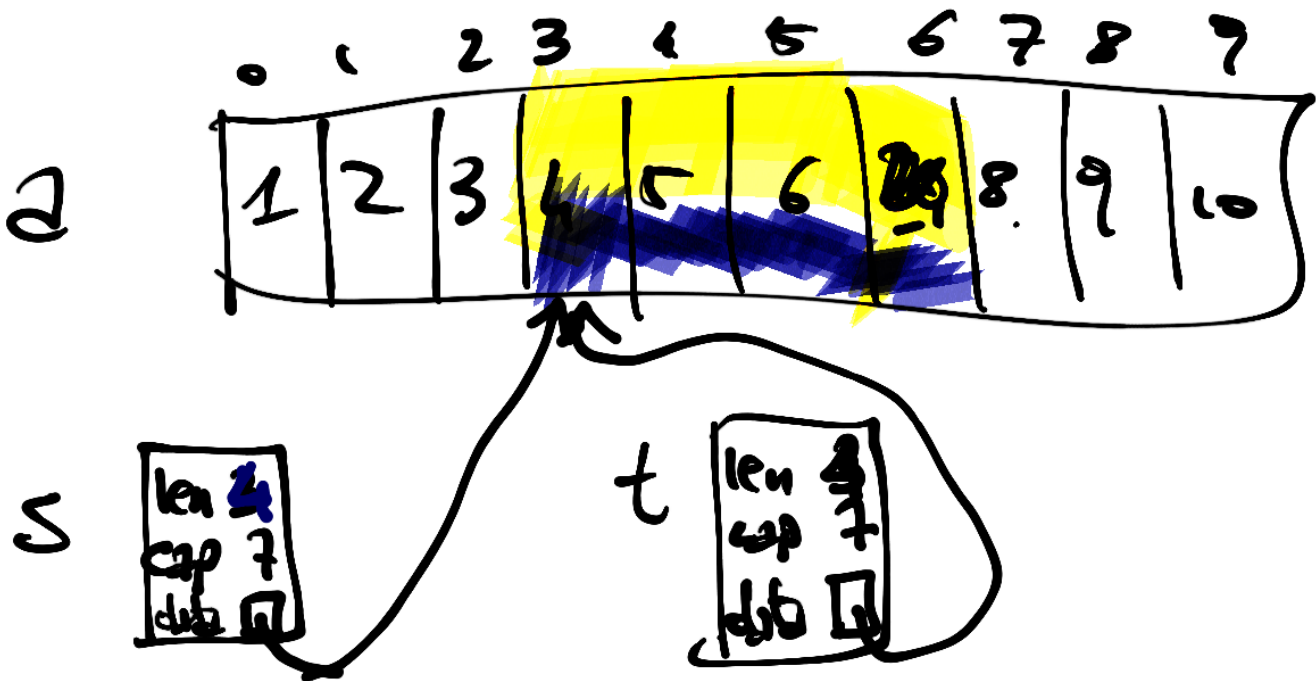
var a [10] int = [10] int {1, 2, 3, 4, 5, ..., 10}

s = a [3:6] ←

t = s

s = append(s, 20)

t = append(t, -1)



var a [10] int

var s [] int

s = a [3:6]

indice in. →

← indice fin. (escluso)

SUBSLICING DI
ARRAY

a [:6]

a [3:]

a [:]

]

```
func media(a []float64) float64 {
```

```
    s := 0.0
```

```
    for _, x := range a {  
        s += x
```

```
    }  
    return s / float64(len(a))
```

```
}
```

```
var pippo [100] float64
```

```
...  
fmt. Println (media (pippo[:]))
```

```
func allOne (a [100] int) {  
  for i:=0; i<len(a); i++ {  
    a[i]=1  
  }  
}
```

allone (Pippo)

```
func allOne (a [] int) {  
  for i:=0; i<len(a); i++ {  
    a[i]=1  
  }  
}
```

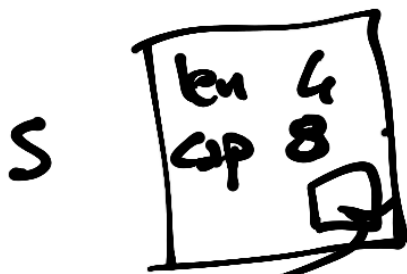
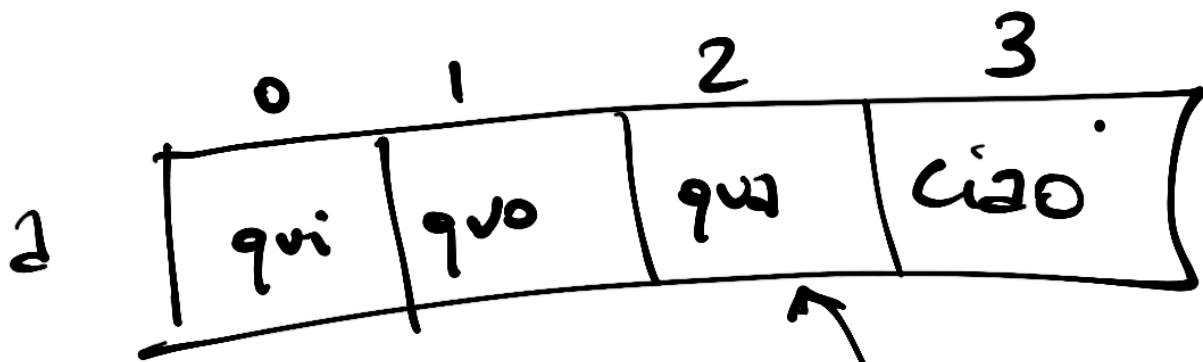
allOne (pippo[:])

```
var a [4]string = {"qui", "quo",  
"qu", "partullo" }
```

```
var s, t []string;
```

```
s = a[1:4]
```

```
t = s[1:2]
```



```
t = append(t, "ciao")
```

```
s = append(s, "addio")
```


- SCRIVETE UNA FUNZIONE
CHE DETERMINA SE UNA
CERTA SLICE DI STRINGHE
CONTIENE UNA CERTA
STRINGA

```
func contains(slice []string, s string) bool {  
    for _, elem := range slice {  
        if elem == s {  
            return true  
        }  
    }  
    return false  
}
```

}

SCRIVETE UNA FONZIONE
CHE DATE DUE SLICE
S E t RESTITUISCA
UNA NUOVA SLICE CON
GLI ELEMENTI DI S
CHE COMPATONO ANCHE
IN t

OUTPUT 1.1

fmt. Printf

fmt. Printf (" STRINGA DI
FORMATO " ,)

↑
VERBI

ARGOMENTI
DI
STAMPA

%v	formato di default
%T	tipo
%d	intero
%g	floating point
%s	stringhe

fmt. Printf ("Ti ho dato %.d lire" , a)

560

↑
int

fmt. Printf ("Il tasso %.g da un valore
di %.d" , x , int64 (int64))

% 5 d

← STARTA VUTERO
CON 5 Litre
(padding con
allin. 2 d x)

% 10 g

% 10.2 g

% [WIDTH] [.PREC]
↳ Verbio

% .2 g