

SLICE

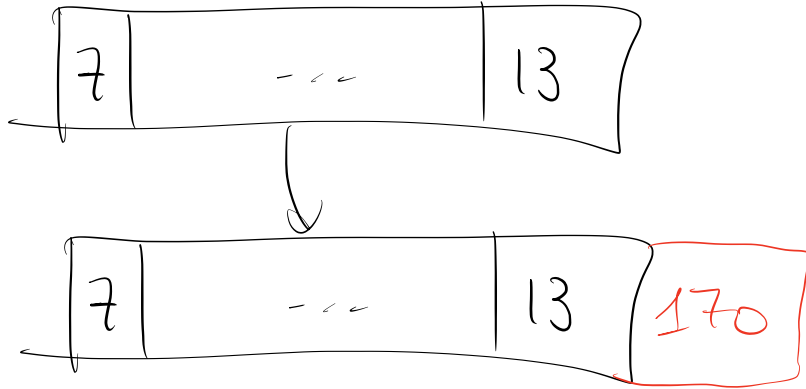
var $x \text{ [] int}$ $x \}$ SLICE
var $a \text{ [10] int}$ $x \}$ ARRAY
var $b \text{ [11] int}$ $x \}$

$x = \text{make}(\text{[] int}, n)$
↙ length

x $\begin{array}{c} 0 \quad 1 \quad \quad \quad n-1 \\ \boxed{0 \mid 0 \mid \dots \mid 0} \end{array}$
 $x[0] \quad x[1] \quad \dots \quad x[n-1]$

- Ⓘ for $i := 0; i < \text{len}(x); i++$
- Ⓜ for $i := \text{range } x$
- Ⓢ for $i, a := \text{range } x$
↙ ↘
INDEX VALUE
 $(x[i])$

x = append(x, 170)

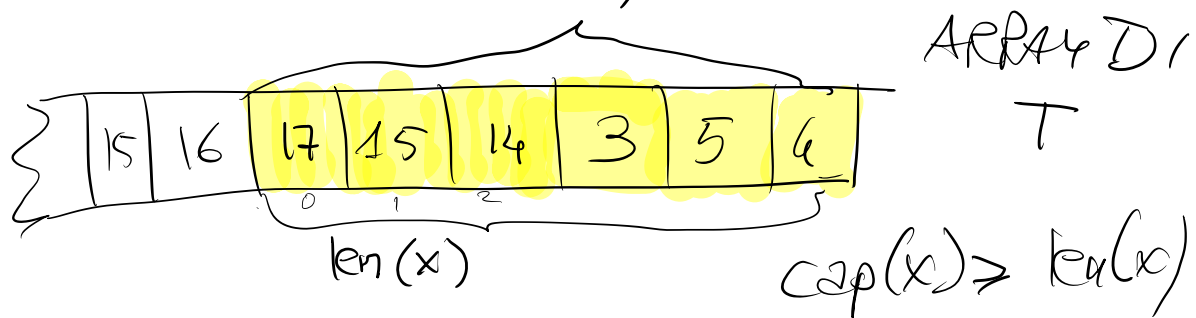
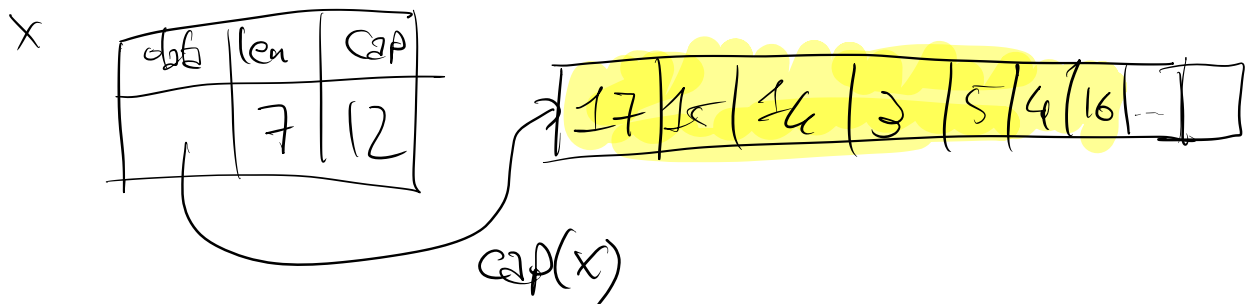


var x [] int

x = append(x, 3)

$\underline{\text{var}} \quad x \quad [] T$
 $\left\{ \begin{array}{l} \underline{\text{var}} \quad x \quad \text{struct} \{ \\ T \quad *data \\ \underline{\text{int}} \quad len \\ \underline{\text{int}} \quad cap \end{array} \right.$

$x = [] \text{int} \{ 17, 30, 14 \}$



$x[1] = 15$

$x = \text{append}(x, 3)$

$x = \text{append}(x, 5)$

$x = \text{append}(x, 4)$

$x = \text{append}(x, 16)$

var x, y $[3]$ int
 $x = [3]$ int $\{3, 7, 2\}$

$y = x$

$x[2] = -1$

x $\begin{array}{c|c|c} 0 & 1 & 2 \\ \hline 3 & 7 & -1 \end{array}$

y $\begin{array}{c|c|c} 0 & 1 & 2 \\ \hline 3 & 7 & 2 \end{array}$

var x, y $[]$ int
 $\rightarrow x = []$ int $\{3, 7, 2\}$

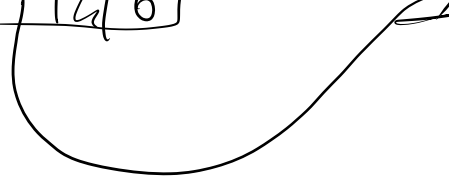
$y = x$

$\begin{array}{|c|c|c|c|c|c|c|} \hline 5 & 7 & -1 & 3 & 1 & -1 & 3 \\ \hline \end{array}$

x $\begin{array}{|c|c|c|} \hline 3 & 7 & 2 \\ \hline \end{array}$

y $\begin{array}{|c|c|c|} \hline 3 & 7 & 2 \\ \hline \end{array}$

$\begin{array}{|c|c|c|c|c|c|} \hline 5 & 7 & -1 & 3 & 1 & -1 \\ \hline \end{array}$



$$x[2] = -1$$

$$x = \text{append}(x, 8)$$

$$x[5] = 5$$

$$x = \text{append}(x, 1)$$

$$x = \text{append}(x, 2)$$

$$x[5] = -1$$

$$y = \text{append}(y, 3)$$

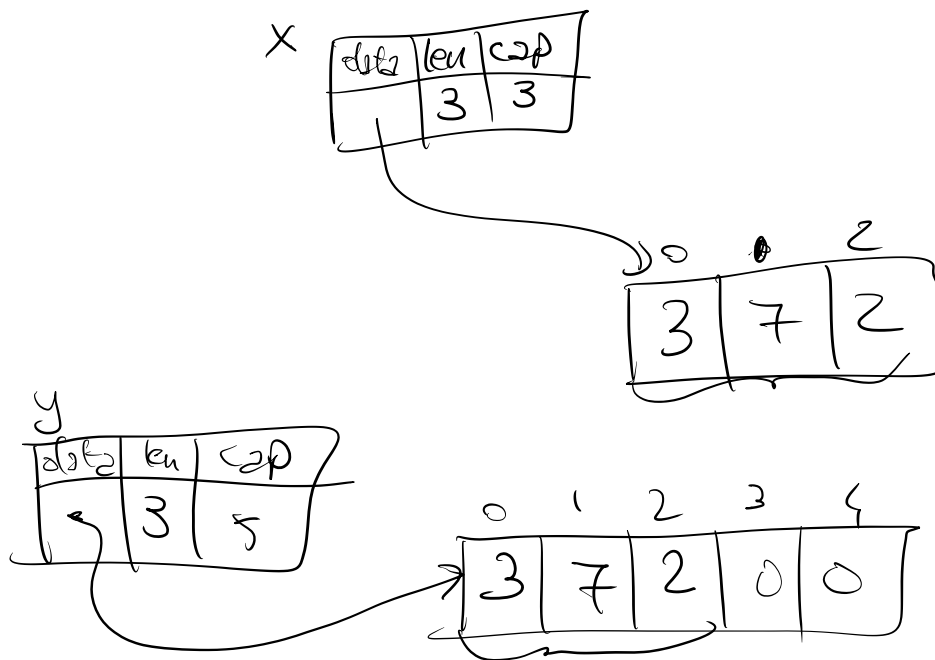
$$x = \text{append}(x, 13)$$

```

var      x, y [ ] int
x = [ ] int {3, 7, 2}

y = make [ ] int, len(x), len(x)+2

for      i := range x {
          y[i] = x[i]
        }
  
```



var x, y [] int

x = make ([] int, 2, 3)

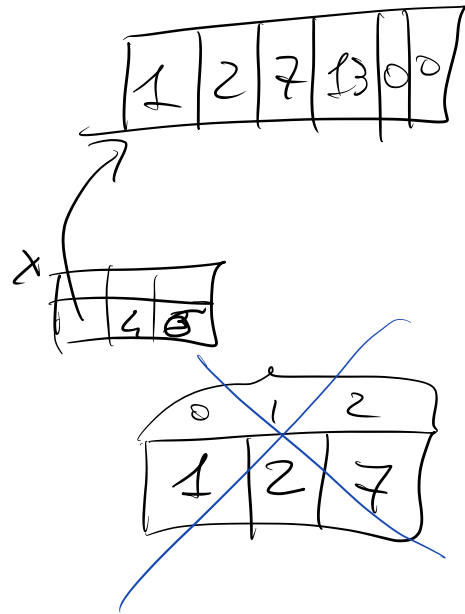
x [0] = 1

x [1] = 2

y = x

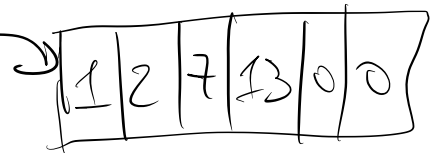
x = append (x, 7)

y = ~~append~~ (y, 7)



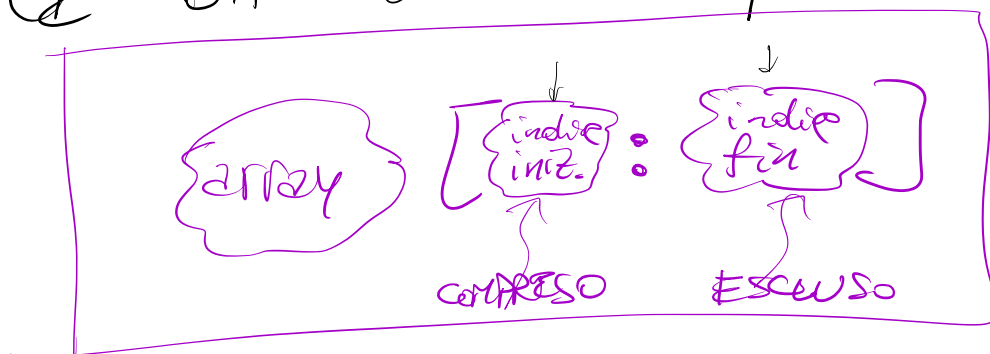
x = ~~append~~ (x, 13)

y = ~~append~~ (y, 13)



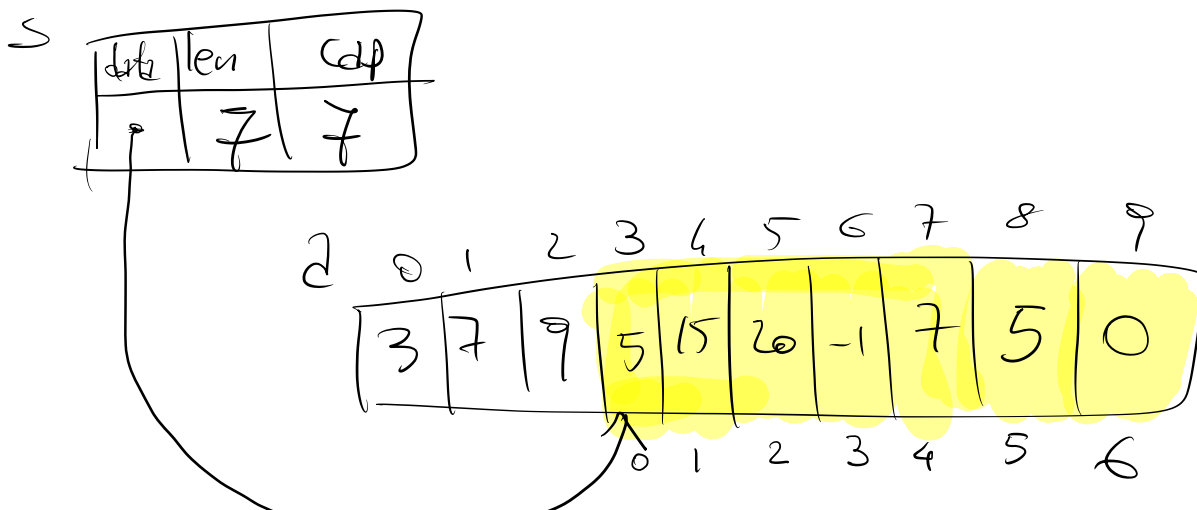
SUBSLICING DI ARRAY

- OPERAZIONE CHE CREA UNA SLICE DA UN ARRAY



$\text{var } a \text{ } [10] \text{int}$
 $a = \dots \text{int } \{3, 7, 9, 11, 15, 20, 11, 7, 9, 11\}$

$\text{var } s \text{ } [] \text{int}$
 $s = a[3:8]$



```
for    - , x := Range s {  
    Fort.Println(x)  
}
```

$S[3] = -1$

$d[3] = 5$

$s = \text{append}(s, 5)$

$s = \text{append}(s, 0)$

FUNCTION MEDIA

```
func media (x [int]) float64 {  
    s := 0.0  
    for -, t := range x {  
        s += float64(t)  
    }  
    return s / float64(len(x))  
}
```

```
...  
var s [int] {3, 7, 9}  
media (s)
```

```
var a [5] int {1, 2, 3, 4, 5}  
a = [...] int {1, 2, 3, 4, 5}  
media (a[:])  
a
```

media (2 [1:4])

SUBSLICEN L DI SLICE

slice [{ ind. inizi. } : { ind. fin. }]

- CREA UNA SLICE CHE HA LO STESSO ARRAY SOGGACENTE DELLA SLICE DATA

$a := [\dots] \text{int } \{1, 2, 3, 4, 5\}$

$s = a[1:3]$

$s[0] = -1$

$s = \text{append}(s, -2)$

$s = \text{append}(s, -3)$

$s = \text{append}(s, -4)$

$\rightarrow s = \text{append}(s, -5)$

$s[0] = -2$

a	0	1	2	3	4
	1	-1	3	-2	-3

s	data	len	cap
		6	8

s	-2	3	-2	-3	-4	-5	0	0
-----	----	---	----	----	----	----	---	---

2 contiene 1, -1, 3, -2, -3

5 contiene -2, 3, -2, -3, -4, -5

ESERCIZI

- FUNZIONE CHE DATA UNA
SLICE DI int E DATO
UN int, DETERMINA SE
LA SLICE LO CONTIENE

func contains (haystack []int,
 needle int) bool {

 for _, t := range haystack {

 if t == needle {

return true

 }

}

 return false

RETURN 1000

}

PER CBT

- IMPLEMENTARE UN main()
- PER PROVARE CHE STA FUNZ.
- IMPLEMENTARE UNA FUNZIONE
- SIMILE CHE CONTA LE
- OCCORRENZE

- DARE DUE SLICE X E Y, RESTITUISCI UNA NUOVA SLICE CHE CONTIENE GLI ELEMENTI DI X CHE COMPARONO IN Y

```
func project (x [int], y [int]) [int]
    var res [int]
    → for _, t := range x {
        if contains(y, t) {
            res = append(res, t)
        }
    }
    return res
}
```

- SCRIVERE UN MAIN

```

func project (x C[int], y C[int]/[st])
var res C[int]

for _, t := range x {
    var i int
    for i=0; i<len(y); i++ {
        if y[i] == t {
            break
        }
    }
    if i<len(y) {
        res = append(res, t)
    }
}

return res
}

```


- SCRIVI UNA FUNZIONE CHE
DATA UNA SLICE DI
DATE, RESTITUISCE LA
PRIMA (in ordine cronologico)

↑
~~FUNZIONE~~ CHE DATA DUE
DATE d_1, d_2 DETERMINA
SE $d_1 \leq d_2$