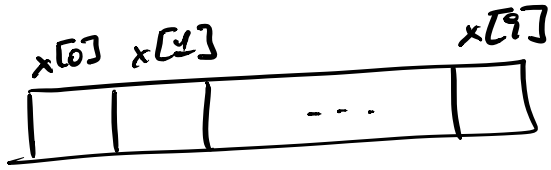
ARRAY

var x, z [10] int

var y [9] int

tipo diverso

x



len(x)

x = z ⇐ COPIA

if x == z ⇐ CONFRONTO (ELEM. PER ELEM.)

func f(x [10]int) ...

f(z)

LETTERALI ARRAY

x := [8]int {0, 1, -1, 4, 2, 2, 3, 3}

x := [8]int {0, 1, 2} //resto = 0

x := [...]int {0, 1, 2, 3} // array lungo 4

x := [8]int {1: 3, 6: -1}

indice indice

```
for  i:=0; i<len(x); i++ {
        ...
}
```

I°

```
for  i := range x {
        ...
}
```

II°

indice

valore

```
for  i, a := range x {
        ...
}
```

x[i]

a

III°

(in lettura)

SLICE

[ ] T

tipo di base

no dimensione

dichiarazione

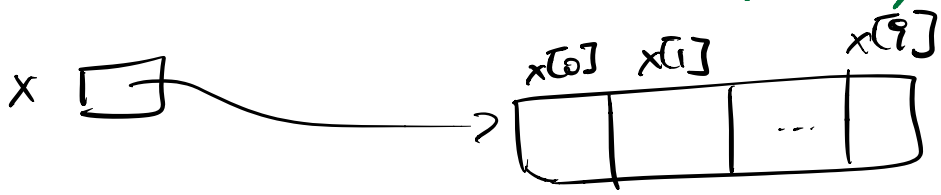
var x []int

creazione

x = make([]int, 10)

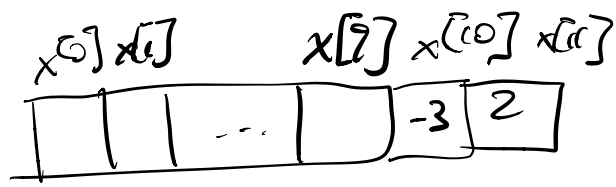
tipo della slice

lunghezza (può essere una qualunque espressione intera)



x = append(x, -3, 2)

prolungamento della slice



LETTERALI SLICE

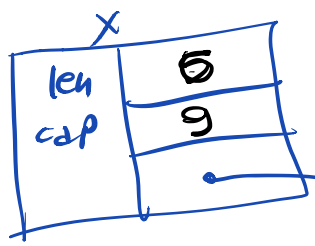
```
x := []int {1, 3, 4}
x := []int {}
```

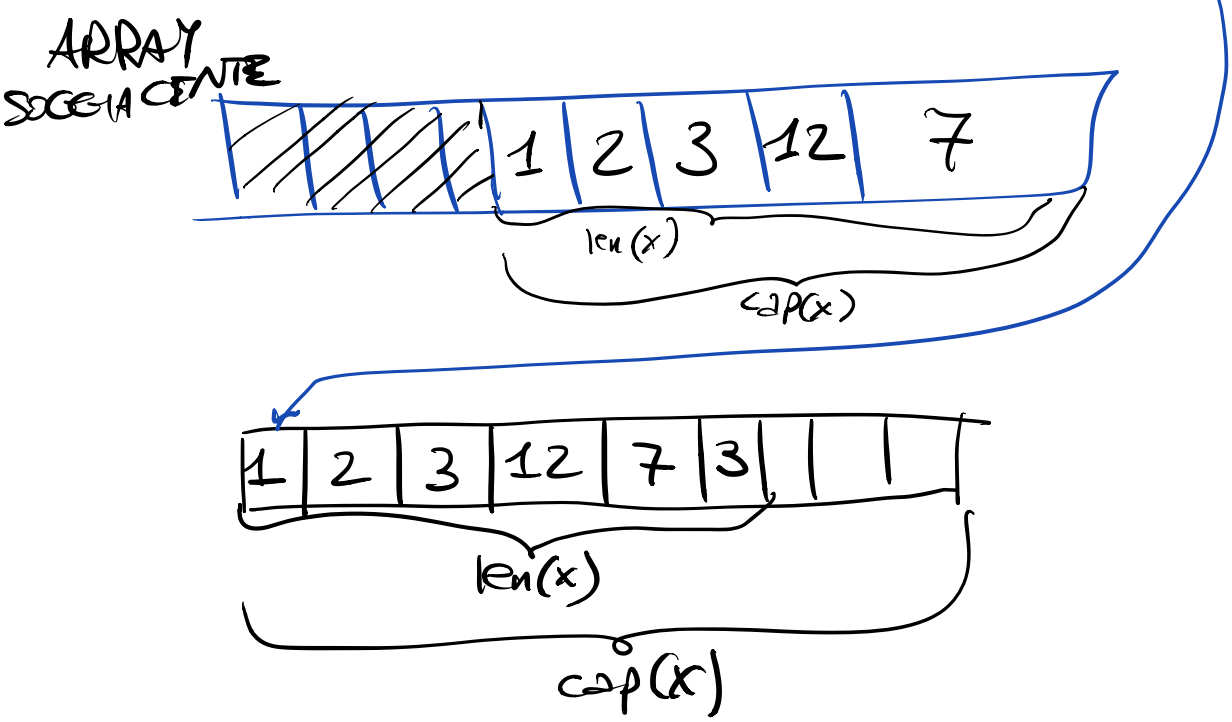
LE SLICE: DIETRO LE QUINTE

```
var x []int
x = []int{1,2,3}
```

```
x struct {
    len, cap int
    data uintptr
}
```

```
x = append(x, 12)
x = append(x, 7)
x = append(x, 3)
```

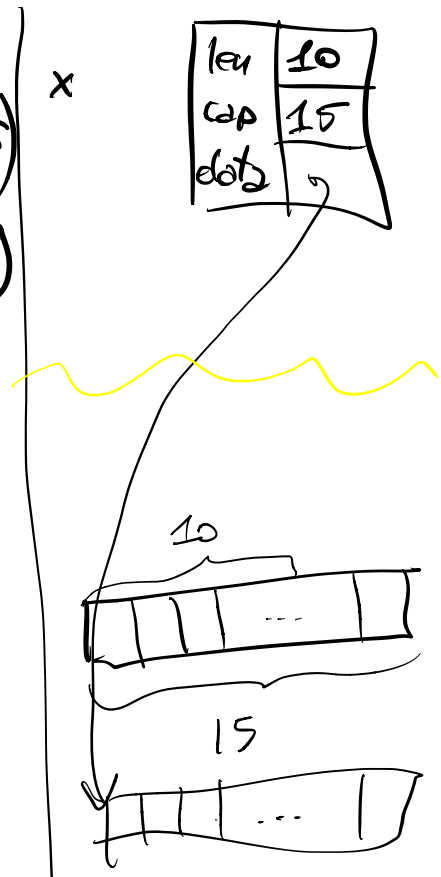




var x []int

x = make([]int, 10, 15)

x = make([]int, 10, 15)



SUBSLICING

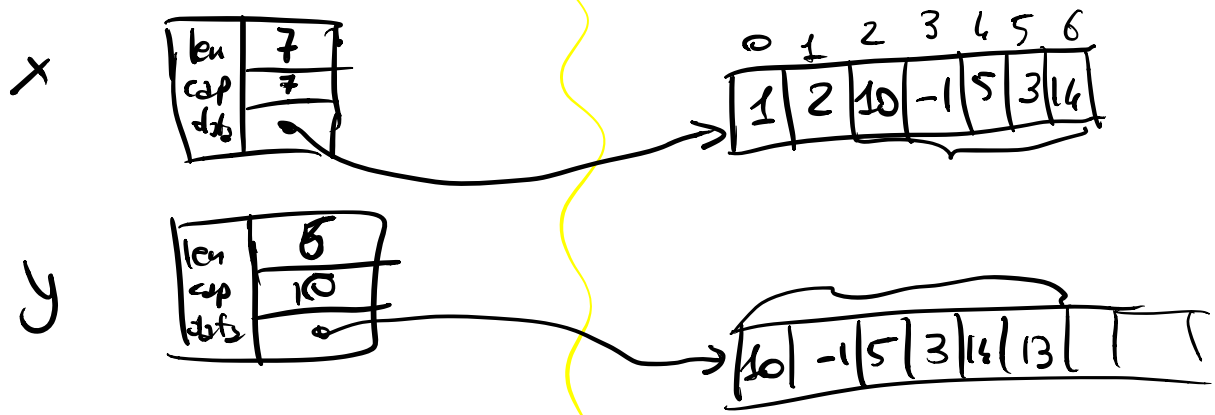
var x, y []int

→ x = []int {1, 2, 3, 4, 5, 6, 7}

→ y = x[2 : 5] ←

indice iniziale (compreso)

indice finale (escluso)



ALIASING!

x[3] = -1

y[0] = 10

y = append(y, 3)

y = append(y, 14)

y = append(y, 13)

---

var x, y []int

x = []int {1, 2, 3, 4, 5}

y = x | y = x[:]

ESERCIZIO

- Scrivere un programma con una funzione che decide se un numero è primo.
Nel main, chiedere all'utente un valore n e inserire in una slice tutti i primi <= n.
Stampare la slice, la sua lunghezza e la sua capacità.

(Prima di eseguirlo, tentate di indovinare quale sarà la capacità)