

$x_1, x_2, \dots, x_n \in \mathbb{R}$

VETTORI (ARRAY E SLICE)

```
func main() {  
    var n, s, x int  
    fut. Scan (&n)  
    for i := 0; i < n; i++ {  
        fut. Scan (&x)  
        s += x  
    }  
    media := float64(s) / float64(n)  
    fut. Printf (media)  
}
```

x_1, x_2, \dots, x_n

$$\mu = \frac{\sum_{i=1}^n x_i}{n}$$

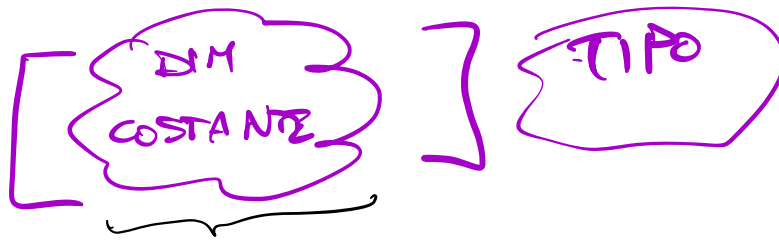
media

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n}}$$

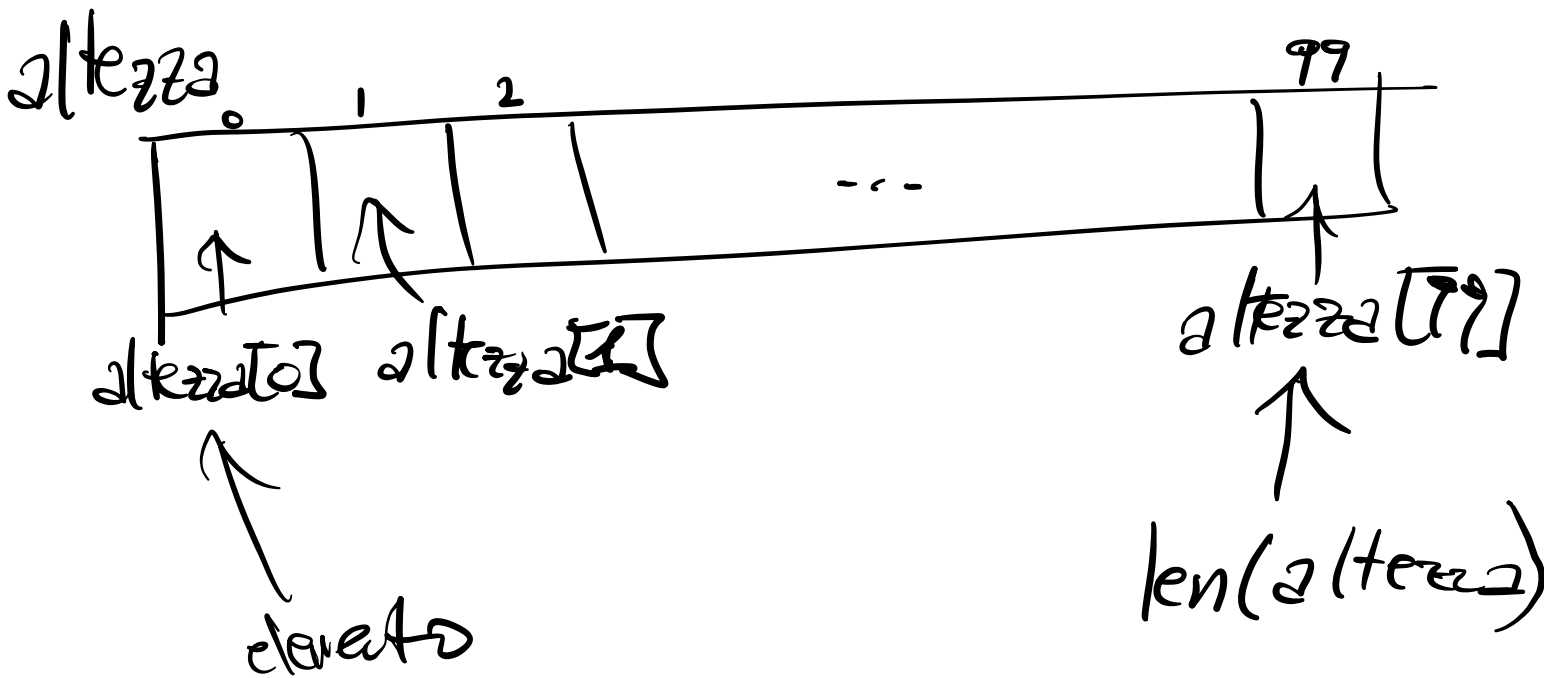
scarto
quadrato
medio

ARRAY

TIPO GIÀ
DEPOSITO



```
var altezza [100] int  
var x [65] string  
var data [1000] data
```



func

```
main() {
  var n int
```

```
  fut. Scan(&n)
```

```
  var a[200] int
```

POPOLAZIONE

```
  for i:=0; i<n; i++ {
    fut. Scan(&a[i])
  }
```

CALCOLO MEDIA

```
  s:=0
  for i:=0; i<n; i++ {
    s+=a[i]
  }
```

```
  media := float64(s) / float64(n)
```

CALCOLO SCARTO Q. MEDIO

```
  sq:=0.0
  for i:=0; i<n; i++ {
    sq+= (a[i]-media)*(a[i]-media)
  }
```

```
  sqm := math.Sqrt(sq / float64(n))
```

SCANSIONE DI ARRAY

I°

```
for i := 0; i < len(a) ; i++ {  
    ... a[i] ...  
}
```

len(a)
n

II°

```
for i := range a {  
    ... a[i] ...  
}
```

finis alla lunghezza dell'array

III°

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

INDICE ELEMENTO CORR.

for $i := 0; i < n; i++ \{$
 $sq += (a[i] - media) *$
 $(a[i] - media)$

}

FOR-RANGE
SEARCHING
(INTERAMENTE)



~~for $x := range a \{$
 $sq += (x - media) *$
 $(x - media)$~~

~~}~~

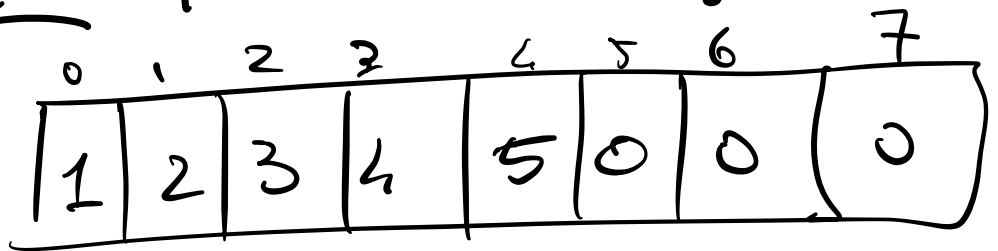
for $i := 0; i < n; i++$ }
 $S += a[i]$
}



for $-, x := \underline{\text{range } a}$ }
 $S += x$
}

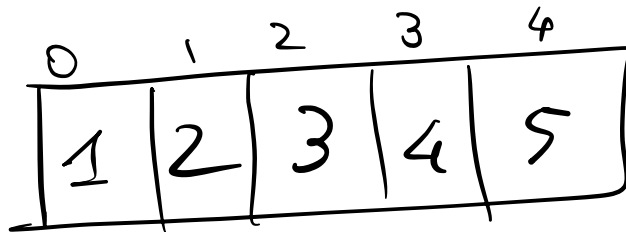
LETTERACI ARRAY

• $[8] \underline{\text{int}} \{1, 2, 3, 4, 5\}$



Es. $x := [8] \underline{\text{int}} \{1, 2, 3, 4, 5\}$

• $[...] \underline{\text{int}} \{1, 2, 3, 4, 5\}$



Es. $x := [...] \underline{\text{int}} \{1, 2, 3, 4, 5\}$

• $[40] \underline{\text{string}} \{1: "ciao", 7: "vadaa" \}$

Es.

$x := [40] \underline{\text{string}}$ ~~~~~

Var x, y [8]int
Var z [7]int

x = y

← copia del
contenido

~~z = x~~

~~x = z~~

func f (a [10]int) }

}

SLICE

[] TIPO

<u>var</u>	x	[] <u>int</u>	} len(—)
<u>var</u>	a	[] <u>string</u>	
<u>var</u>	b	[] <u>data</u>	

↓
LUNGHE. ZERO

CREAZIONE

x = make ([] int, 100)
a = make ([] string, (n+1)/4)
b = make ([] data, 7)

func

main () {

var n int

fmt. Scan (&n)

POPOLAZIONE

var a [] int (int, n)

for i := 0; i < n; i++

fmt. Scan (&a[i])

}

CALCO
MEDIA

s := 0

for

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

}

media := float64(s) / float64(n)

CALCO
SCARTO
Q. MEDIO

sq := 0.0

for

_, x := range a {
sq += (x - media) *
(x - media)

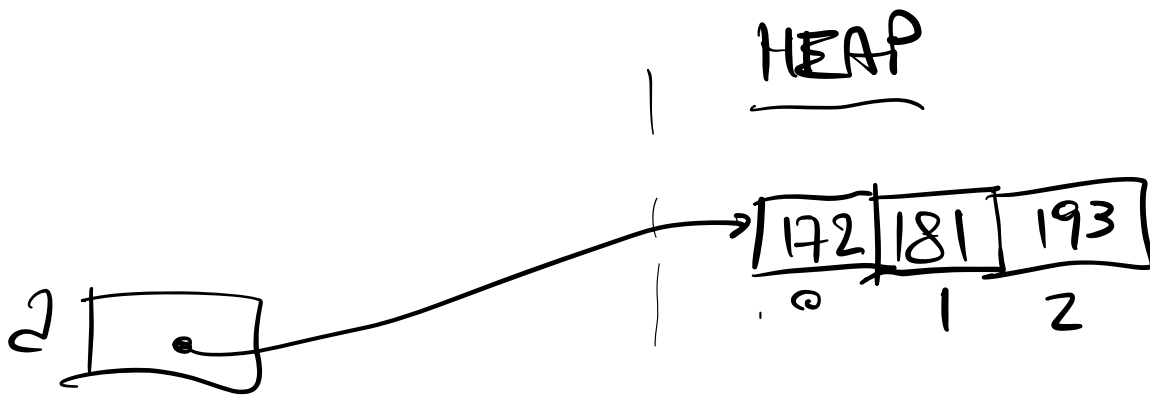
}

sqm := math.Sqrt(sq / float64(n))

APPEND

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

$x = \text{append}(x, v_1, v_2, \dots)$



172
181
193
0

func

main ()
var n int

var a [] int

for

var x int

fmt.Scan (&x)

if x == 0 {
break

}

a = append (a, x)

}

s := 0

for

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

}

media := float64 (s) / float64 (n)

sq := 0.0

for

_, x := range a {
sq += (x - media) *

Ardo
MEDIA

Ardo

Q1
SEARTO
Q.MEDIO

$$sq += \begin{pmatrix} x - \text{media} \\ x - \text{media} \end{pmatrix}$$

$$\} \\ sqm := \text{math.sqrt}(sq / (n - 1))$$

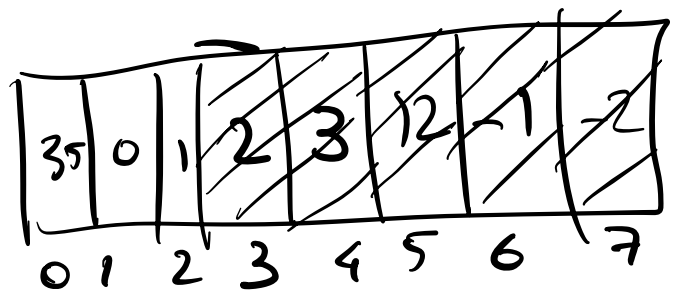
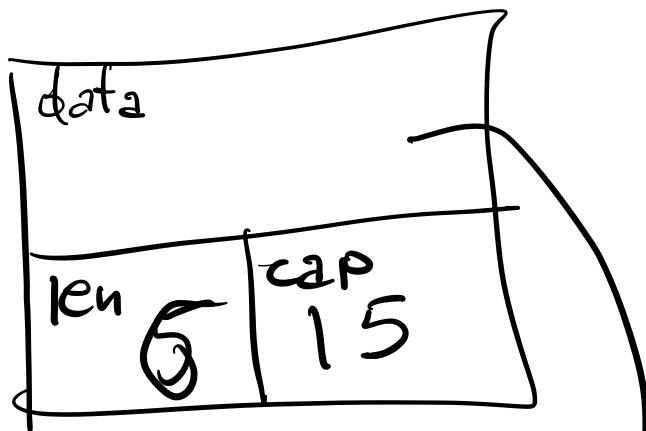
LETTERAZI SLICE

[] int { 1, 3, 5 }

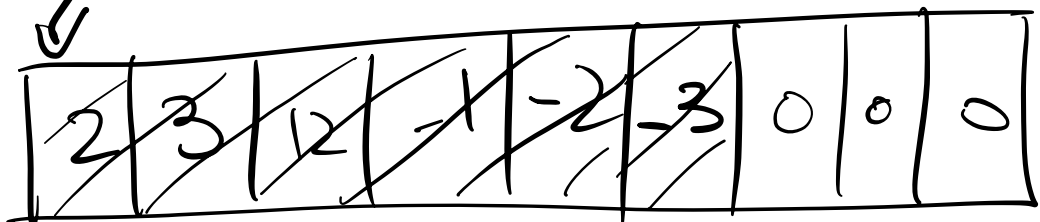
[] string { "ciao", "pippo" }

DIETRO LE QUINTE

2 [] int



HEAD

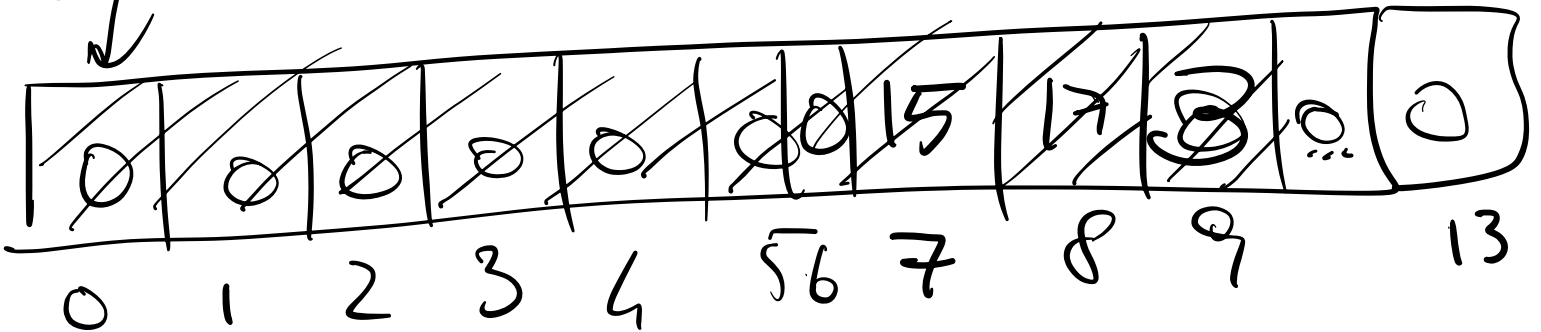
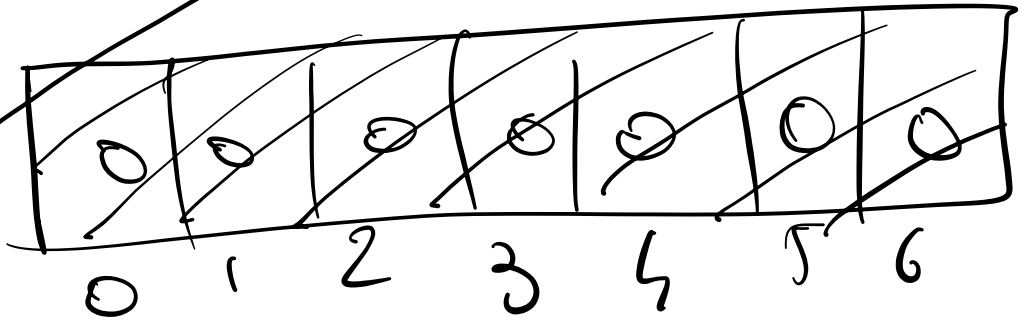
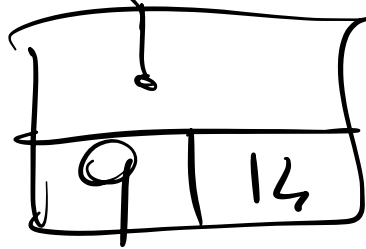
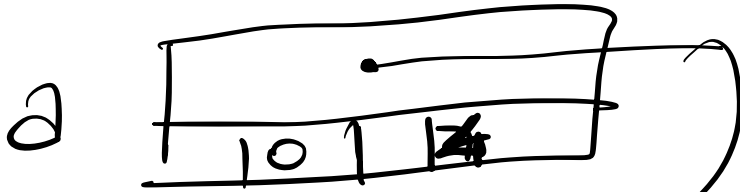


Var a, b [int]

a = make([int, 7])

a = append(a, 15) ←

a = append(a, 17)



$$b = a$$

$$a = \text{append}(a, 6)$$

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

var minimo string

primo Volta = true

scanner := bufio.NewScanner(
os.Stdin)

scanner.Split(bufio.ScanWords)

for scanner.Scan() {

word := scanner.Text()

~~primo Volta~~ if word < minimo {

minimo = word

primo Volta = false

}

)

}