

ARRAY E SLICE

$$x_0, x_1, \dots, x_{n-1} \in \mathbb{R}$$

$$\frac{\sum_{i=0}^{n-1} x_i}{n}$$



ARRAY

[dim costante] T

var x [10] int

var y [5] string

var z [30] persona

var w [5] [10] int

ACCESSO

x [i]

espr. intera
 $i = 0, \dots, \text{len}(x) - 1$

ESEMPIO VARIANZA

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

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

var alt [200] int
var n int

```
for n < len (alt) {  
    fut. Scan (&h)  
    if h == 0 {  
        break  
    }  
    alt[n] = h  
    n++  
}
```

POPOLA
L'ARRAY

```

media := 0.0
for i := 0; i < n; i++ {
    media += float64(alt[i])
}
media /= float64(n)

```

```

varianza := 0.0
for i := 0; i < n; i++ {
    varianza += (float64(alt[i]) - media) *
                (float64(alt[i]) - media)
}
varianza /= float64(n)

```

CODE SUGLI ARRAY

Var x, y [10] persona

...

x = y

if

x == y

/ x != y

Var z [10] int

Print {1, 7, 2, 4, 3, 5, 2, 0, 1, 3}

$z = [10] \text{ int } \{1, 7, 3, 4, 2\}$

$z = [10] \text{ int } \{1, 7, 3, 4\}$

$w := [\dots] \text{ int } \{1, 7, 4, 2\}$

letterali di
topo array

FOR

E SLICE

PER ARRAY

I°

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

II°

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

III

```

    }
    for i, v := range a {
        ...
        a[i]
        ...
    }

```

INDICE (pointing to *i*)

VALORE (pointing to *v*)

MEDIA

```

    media := 0.0
    for i := 0; i < n; i++ {
        media += float64(a[i])
    }
    media /= float64(n)

```

MEDIA

```

    media := 0.0
    for i := range a {
        media += float64(a[i])
    }
    media /= float64(n)

```

MEDIA

```

    media := 0.0
    for _, v := range a {
        media += float64(v)
    }
    media /= float64(n)

```

SLICE

- Superficie alzata: è un array di lunghezza dinamica (solo allungandosi)
- In ogni istante ha una certa lunghezza $len(x)$
- $x[0], x[1], \dots, x[n-1]$
 - $\downarrow$
length.

DICHIARAZIONE

[] T

var x []int

CREAZIONE

make([]T, lengths)

$$\left[\begin{array}{l} \text{var} \\ x = \text{make}([] \text{int}, 100) \end{array} \right]$$

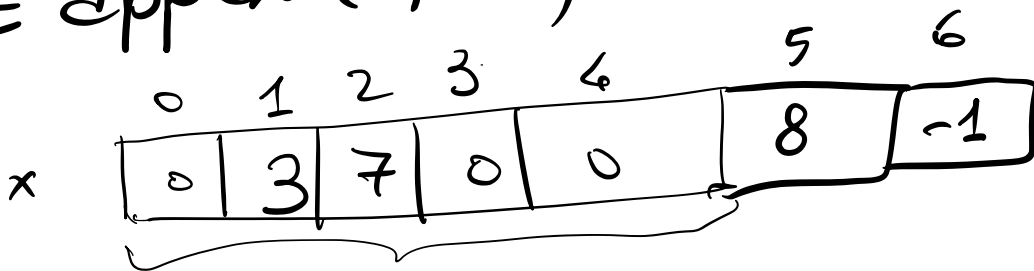
expression
 (non-acc. constant)

	0	1	
x			0

ALLUNCA MENTO

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

```
var x [3int]
x = make([3int, 5])
x[2] = 7
x[1] = 3
x = append(x, 8)
x = append(x, -1)
```



LETTERAZI SLICE

```
z := [3int {1, 3, 7}]
```

var alt int int
var n int

for {
 funt. Scan (&h)
if h == 0 {
 break
 }
 alt = append (alt, h)
 }
 }
 n = len (alt)

POPOLA L'ARRAY

MEDIA [for —, v := range alt {
 media += float64 (v)
 }
 media /= float64 (n)

VAR. [for —, v := range alt {
 varianza += (float64 (v) - media) * (float64 (v) - media)
 }
 varianza /= float64 (n)

ESERCIZI

1) Leggere delle righe (da standard input o da un file) e stamparle dall'ultima alla prima

```
$> go build pippo.go
```

```
$> ./pippo
```

```
~~~~~
```

```
$> ./pippo <dati.txt  
          >out.txt
```

2) con (r) ma ogni riga viene stampata al contrario

3) chiede n, legge n righe e poi da lì in avanti un pattern e stampa le righe quante righe contengono il pattern; smette Ctrl-D

Quante righe? 3

ciao mamma
amore amore mio
ciao bello amato

am

am è contenuto in 3 righe

ao

ao è contenuto in 1 riga

io

io è contenuto in 2 righe

e

^D

ESERCIZIO

Leggi righe e stampa le cartelle