

SUBSLICING & ASSEGNAZIONE DI SLICE

var x, y [10]int

x = y ←
~

if x == y
...

var x, y [3]int

x = y

~~x == y~~

x := [3]int {10, 20, 30, 40}

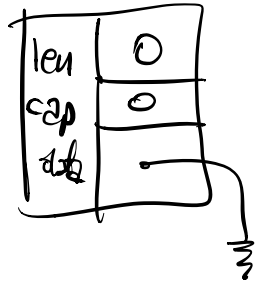
var y [3]int

y = make([3]int, len(x))

copy(y, x)
↑ ↑
destinat. sorgente

make ([3]int, 0, int(float64(2)) /

var x [3]int



SUBSLICING DI SLICE

x := []string {"lon", "var", "wer",
"gio", "ver", "sob", "don"}

y := x[:5]

z := x[5:]

SUBSLICING DI ARRAY

x := [5]float64 {1.0, 1.1, 1.2, 1.3,
1.4}

→ y := x[1:4]
→ y = append(y, 7)
y = append(y, 3.5)

y := x[:]

STACK

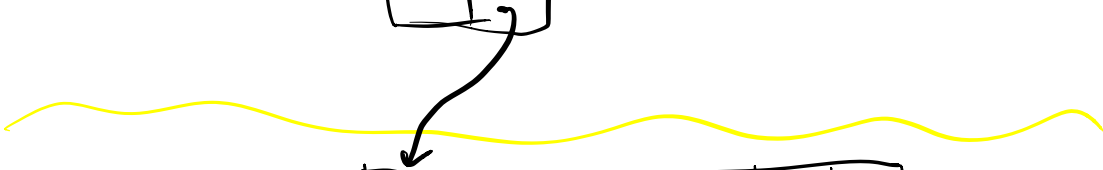
x

1.0	1.1	1.2	1.3	7
-----	-----	-----	-----	---

y

len	5
cap	8

HEAP



1.1	1.2	1.3	7	3.5	0	0	0
-----	-----	-----	---	-----	---	---	---

func f (x [100]int) ...

func f (x [int])

(A)VERBI PER PRINTF

%v

formato di default

%d

intero (decimale)

%b

intero (binario)

%s

stringa

%f

float

%T

tipo

...

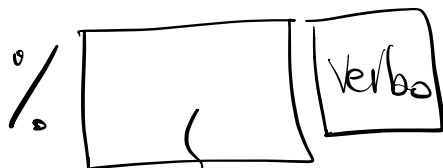
%10d

%10s

%-10d

%-10s

%010d



AVERBI

[WIDTH]. [PRECISION]

lunghezza
minima
complessiva
(in caratteri)

numero di
cifre dopo
la virgola
(%f)

ESERCIZI

1) Scrivere un programma calc.go

./calc 12 + 5

17

./calc 13 / 6

2.1666...

+ - x /

2) Scrivere un programma che riceve come argomenti due date e stampa il numero di giorni fra la prima e la seconda

./dist 2/1/1945 12/1/1980

3) Scrivere una funzione che date due slice di interi stabilisce se sono uguali (stessa length e stesso contenuto)

4) Scrivere una funzione che dato una slice x e y

trova quanti elementi di x
compaiono in y