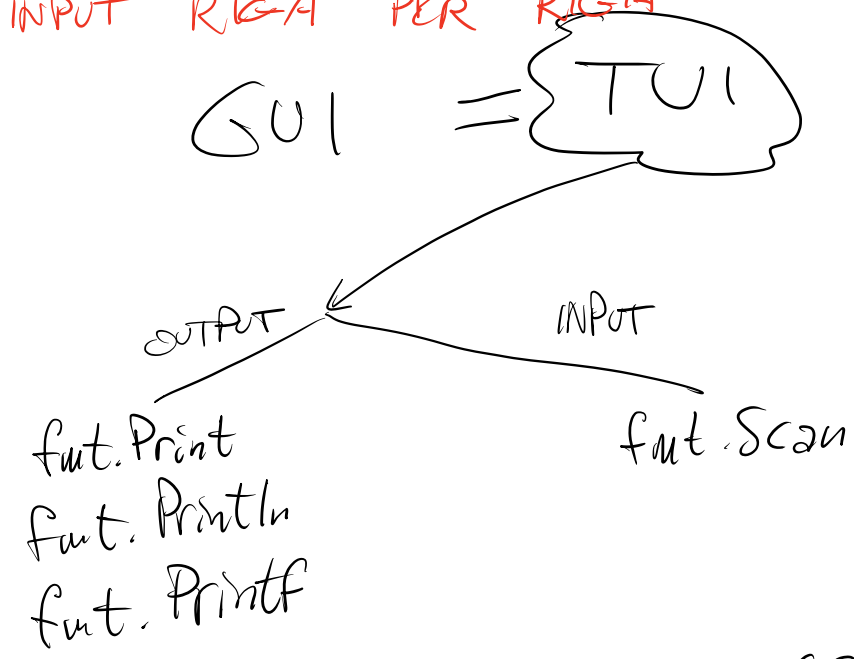
INPUT RIGA PER RIGA



fmt.Scan(&n)

pippo.go

```
import (
    "bufio"
    "os"
)
...
scanner := bufio.NewScanner(os.Stdin)
for scanner.Scan() {
    l := scanner.Text()
    ...
}
```

$> go run pippo.go

Bob è pazzo ↵
ciao come stai ↵
375 ↵

ctrl-D ← EOF

$> go run pippo.go < testo.txt

Bob è pazzo ciao come stai 375

ESERCIZIO

Leggere una poesia e formare la parola costituita dalle iniziali dei versi.

Sempre caro mi fu quest'ermo colle
e questa siepe
che ...

INPUT

Sqec...

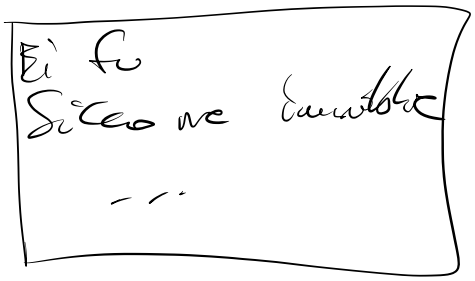
OUTPUT

---

acrostico.go

```
func main() {
    var res string
    scanner := bufio.NewScanner(os.Stdin)
    for scanner.Scan() {
        line := scanner.Text()
        res += string(line[0])
    }
    fmt.Println(res)
}
```

$> go build acrostico.go
$> ./acrostico
Ei fu ↵
Siccome immobile ↵
dato il mortal sospiro ↵
stl- ↵
Esd



CinqueMaggio.txt

$> ./acrostico < CinqueMaggio.txt

ESERCIZIO

ciao come stai

3

o come staicia

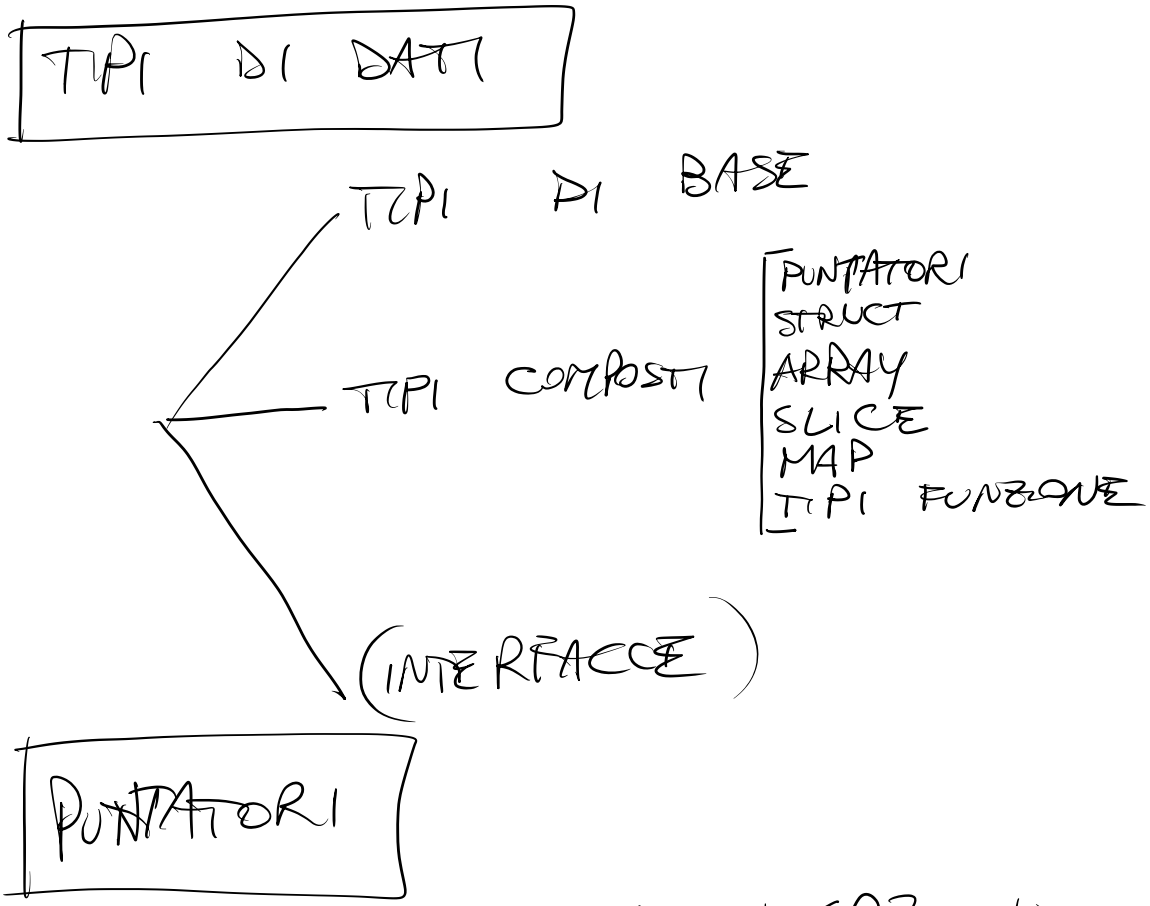
io sto bene e tu

7

bene e tuio sto

...

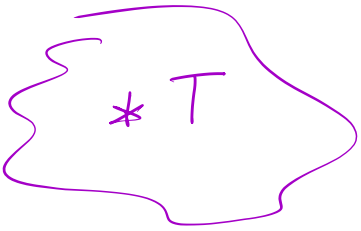
PUNTATORI



PUNTATORI

- RAPPRESENTANO LOCAZIONI DI MEMORIA
- È UNA LOCAZIONE DI MEMORIA A PARTIRE DALLA QUALE SI TROVA UN DATO

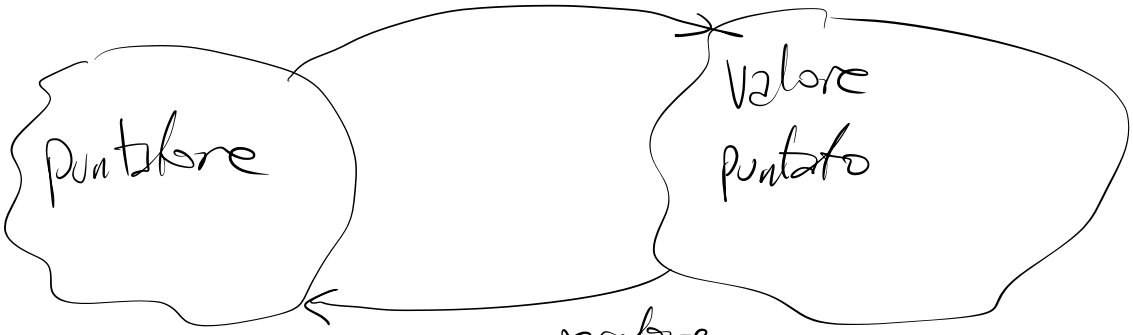
SINTASSI



"puntatore a T"

```
var p *int64
var q *string
var r *bool
```

* operatore di indirezione



& operatore address_of (o di indirizzo)

var x string

x = "ciao"

var p *string ← nil

p = &x

fmt.Println(p)

fmt.Println(*p)

fmt.Println((*p)[0])

fmt.Println(len(*p))

var x int

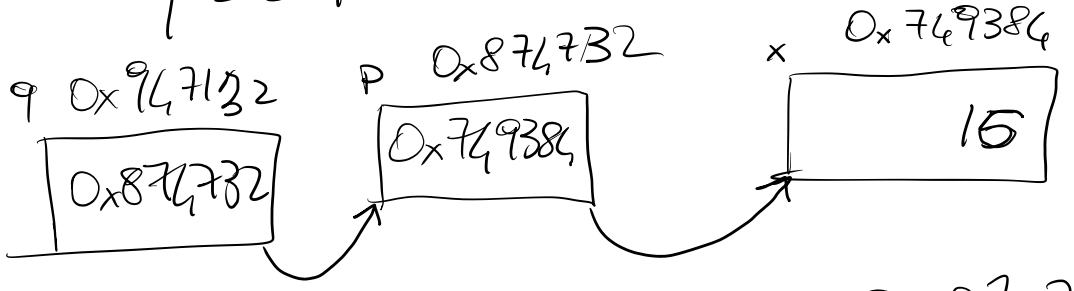
var p *int

var q **int

x = 15

p = &x

q = &p



fmt.Println(q) → 0x874732

fmt.Println(*q) → 0x749384

fmt.Println(**q) → 15

**q++ ≈ x++

fmt.Println(x)

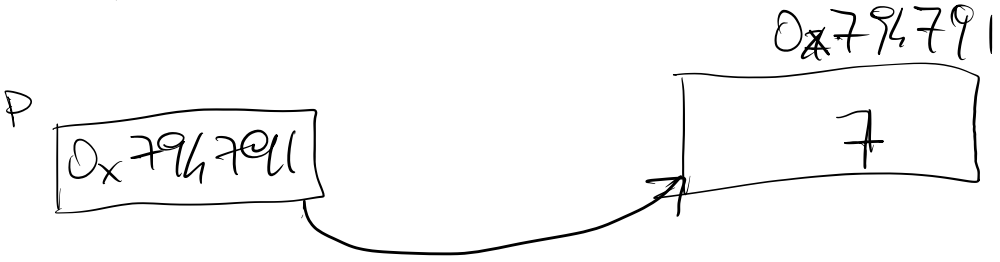
x *p **q

new

new(T)

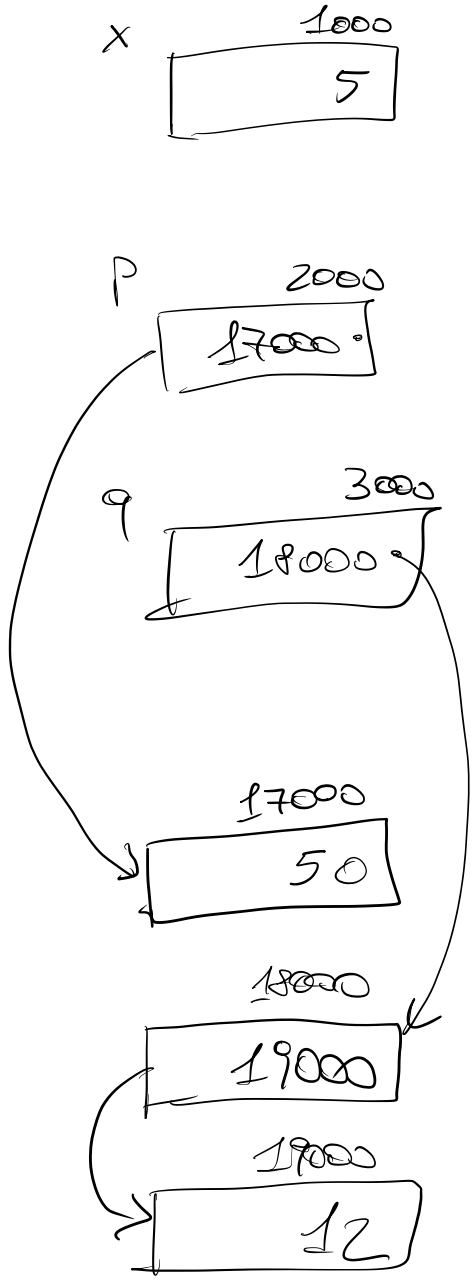
- alloca spazio per T tipo
- restituisce l'indirizzo sotto forma di puntatore

```
var    p   *int64
p = new(int64)
*p = 7
```



var x int
var p *int
var q **int

x = 7
p = &x
q = &p
p = new (int)
*p = 50
q = new (*int)
*q = &x
**q = 5
→ *q = new (int)
**q = 12
fmt.Println(*p + **q)
*p = 2 * (**q)

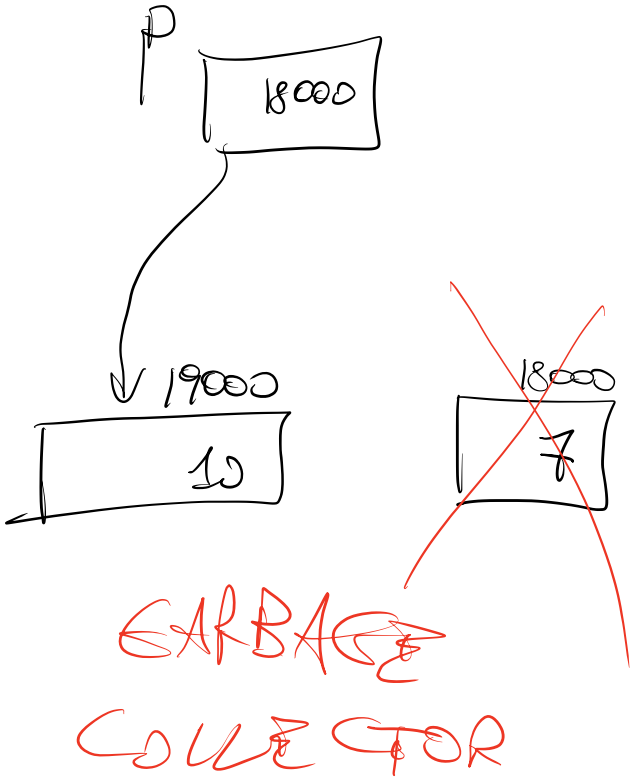


p = new (int)

*p = 7

p = new (int)

*p = 10



GARBAGE COLLECTOR