"7/10/2022" ⟶ 7, 10, 2022

```
func strToDate(x string) (d int, m int, y int, ok bool) {
    var p1 int
    for p1=0; p1<len(x); p1++ {
        if x[p1] == '/' {
            break
        }
    }
    if p1 == len(x) {
        return
    }
    ....
}
```

x[p1+1:]

0	1	2	3	4	5	6
1	2	/	2	0	2	2

p2

x

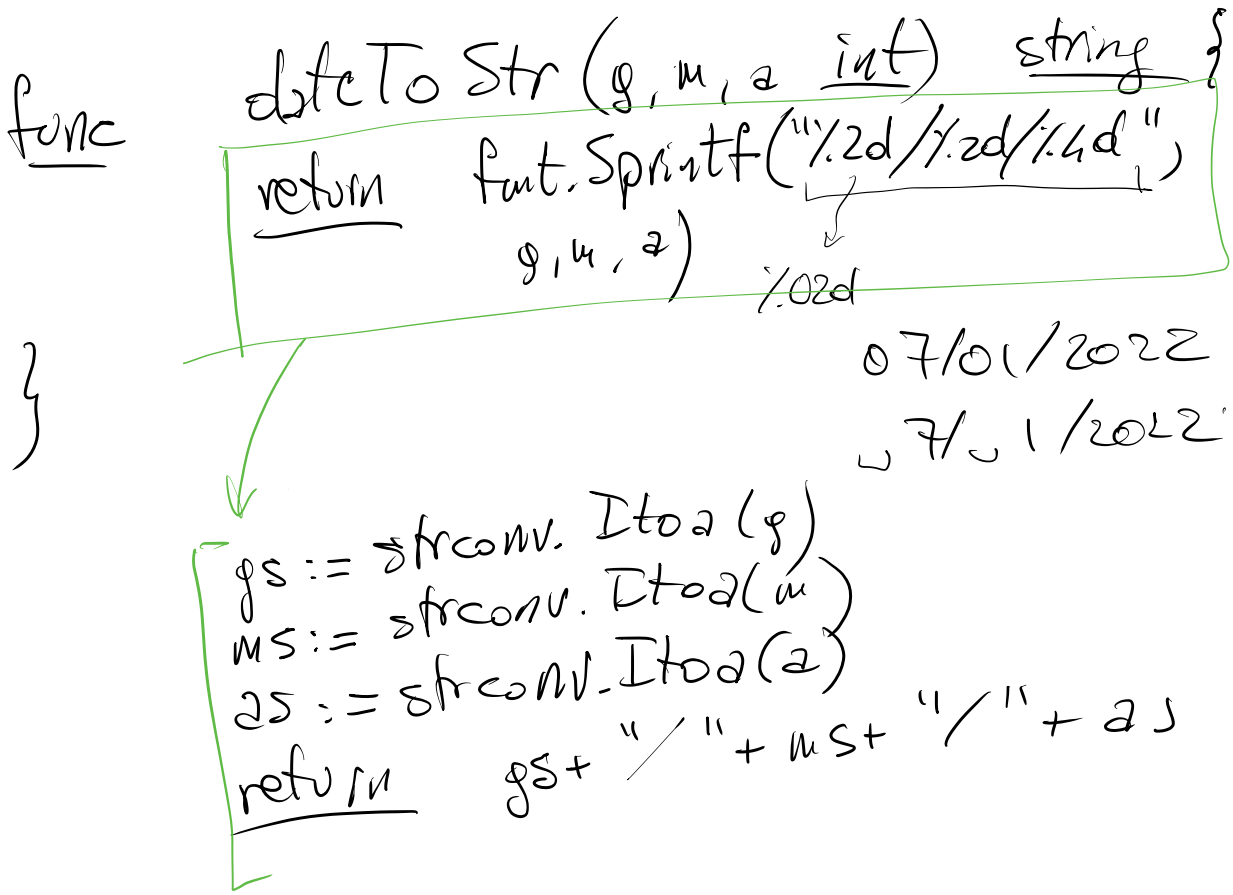
0	1	2	3	4	5	6	7	8	9
3	1	/	1	2	/	2	0	2	2

p1

```
func strToDate(x string) (d int, m int, y int, ok bool) {
    var err error
    p1 := strings.IndexRune(x, '/')
    if p1 < 0 {
        return
    }
    p2 := strings.IndexRune(x[p1+1:], '/')
    if p2 < 0 {
        return
    }
    p2 += p1 + 1
    ds := x[:p1]
    ms := x[p1+1:p2]
    ys := x[p2+1:]
    d, err = strconv.Atoi(ds)
    if err != nil {
        return
    }
    m, err = strconv.Atoi(ms)
    if err != nil {
        return
    }
    y, err = strconv.Atoi(ys)
```

```
if    err != nil {
        return
}
controlli
ok = true
return
}
var    g, m, a   int
var    ok        bool
...
g, m, a, ok = strToDate(x)
if !ok {
    .....
}
```

uso


func
dateToStr(g, m, a int) string {
return fmt.Sprintf("%.2d/%.2d/%.4d",
g, m, a)
%02d
07/01/2022
7/1/2022
}
gs := strconv.Itoa(g)
ms := strconv.Itoa(m)
as := strconv.Itoa(a)
return gs + "/" + ms + "/" + as

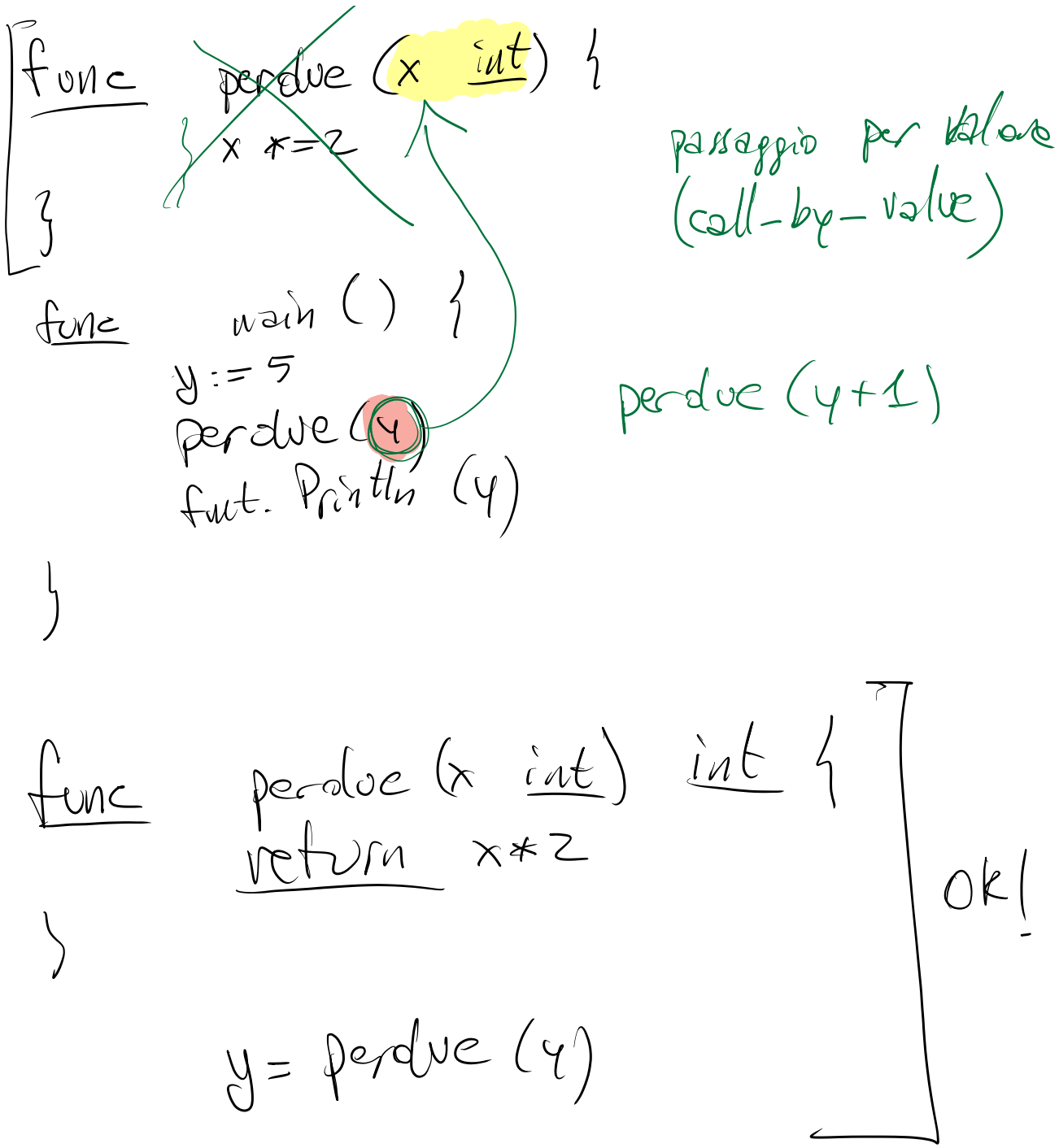
INPUT/OUTPUT

fmt.Scan

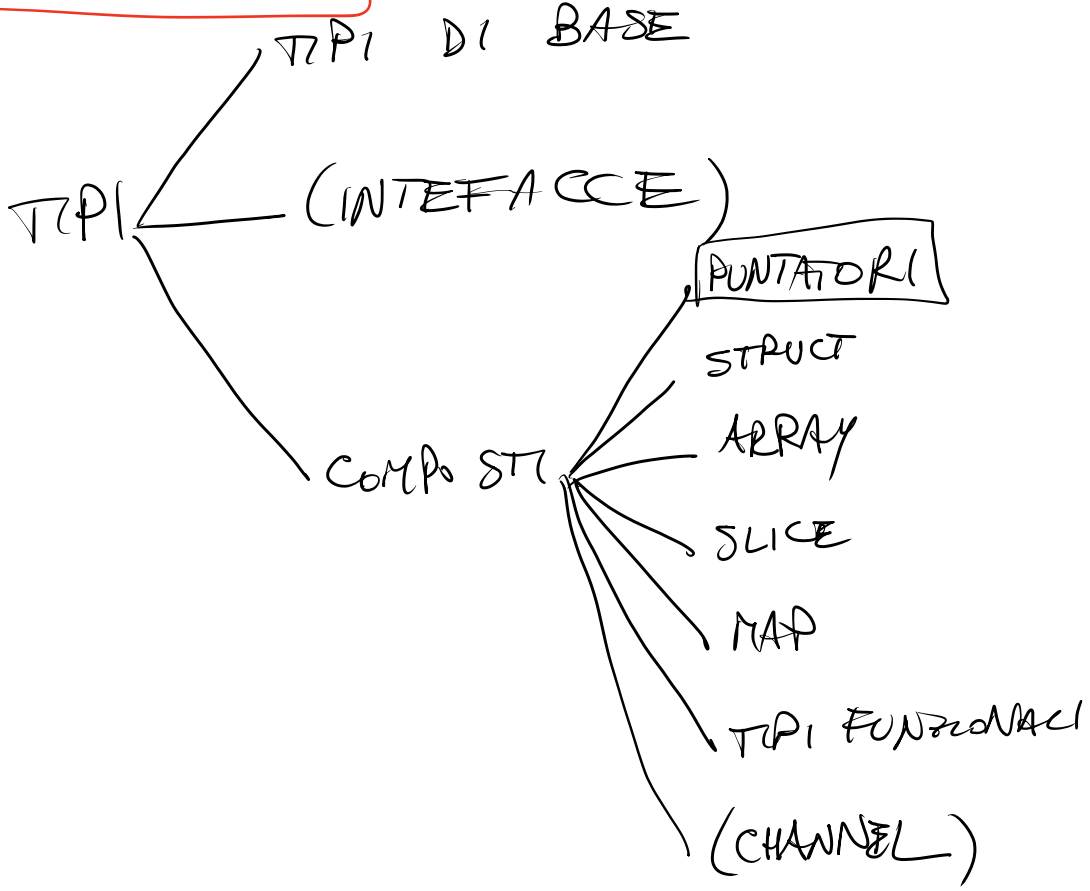
LETTURA RIGA PER RIGA

fmt.Print
fmt.Println
fmt.Printf

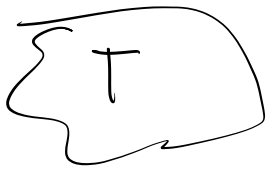
```
import (
    "bufio"
    "os"
)

scanner := bufio.NewScanner(os.Stdin)
for scanner.Scan() {
    line := scanner.Text()
    ... uso line ...
}
```


func perdue (x int) {
x *=2
}
func main () {
y := 5
perdue (y)
fmt. Println (y)
}
passaggio per valore
(call-by-value)
perdue (y+1)
func perdue (x int) int {
return x*2
}
y = perdue (y)
ok!

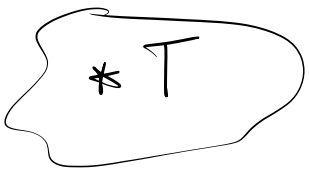
TIPI COMPOSTI



PUNTATORI = indirizzi di memoria (dove si trova un valore)
da dove inizia



TIPO

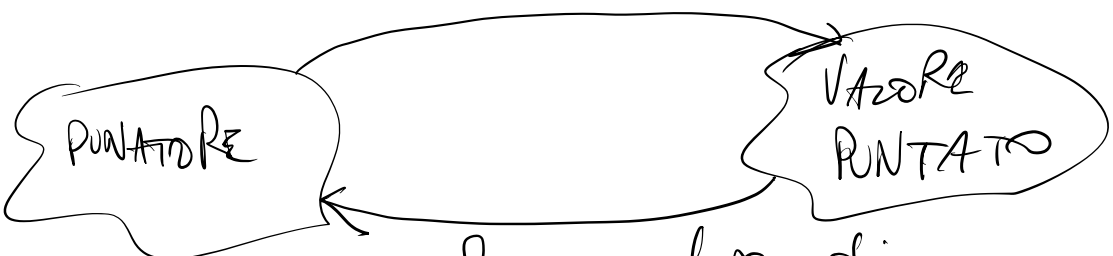


PUNTATORE AL TIPO T

```
var x int
var s string
var f float64
```

```
var p *int
var q *string
var r *float64
```

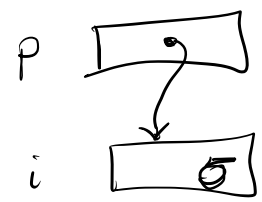
* operatore di indirizzione



& operatore di indirizzo

```
var   x   string
var   p   *string
x = "ciao"
p = &x
fmt.Println(x)          ~> ciao
fmt.Println(p)          ~> 0x3A75F
fmt.Println(*p)         ~> ciao
fmt.Println(string((*p)[0])) ~> c
                           (*p)[0] = x[0]
fmt.Println(len(*p))    ~> 4
```

```
var   i   int
var   p   *int
p = &i
i = 5
(*p)++
fmt.Println(i)
```



var x int

var p *int

var q **int

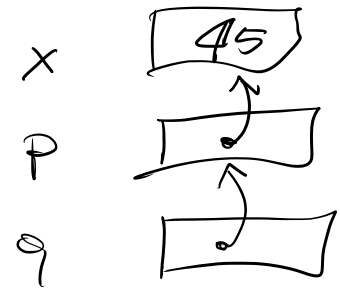
x = 15

p = &x

q = &p

**q += (*p) + x

fmt.Println(x)



var x, y int

var p, q *int

var r **int

x = 7

y = 9

p = &x

q = &y

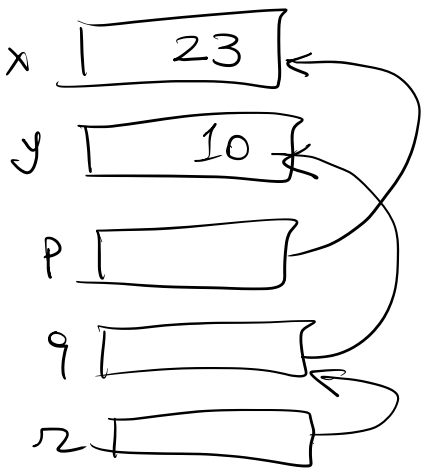
r = &p

**r += *q + x

r = &q

**r++

fmt.Println(x+y) ⟶ 33



```
func perdue(x int) {
    x *= 2
}
```

```
func perdue(p *int) {
    *p *= 2
}

var x int
x = 7
perdue(&x)
```

new

```
new(T)
```

restituisce un *T

```
var p *int
p = new(int)
*p = 7
p = new(int)
*p = 8
```