SELEZIONE MULTIARIA (switch)

espressione

```
switch (selettore) {
case espr, espr, ... :
    A
case espr, espr, ... :
    B
...
default:
    D
}
```

← opz.

- SCRIVERE UNA FUNZIONE CHE DATO UN NUMERO $0 < x < 100$ RESTITUISCA IL NUMERO IN LETTERE

```
func inLettere(x int) string {
    if x >= 10 && x <= 19 {
        switch x {
        case 10: return "dieci"
        case 11: return "undici"
        ...
        case 19: return "diciannove"
        }
        return "CANTHAPPEN"
    } else {
        var dec, uni string
        switch x/10 :
```

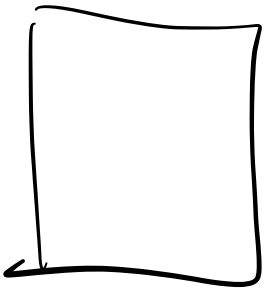
```
switch
case 0:   dec = ""
case 0:   dec = ""
case 2:   dec = "venti"
case 3:   dec = "trenta"
   :
case 9:   dec = "novanta"

}
switch   x%10:
case 0:   uni = ""
case 1:   uni = "uno"
case 2:   uni = "due"
   :
case 9:   uni = "nove"

} x<10 ||
if (x%10==1 && x%10!=8){
      return dec+uni
}else{
      return dec[:len(dec)-1]+
```

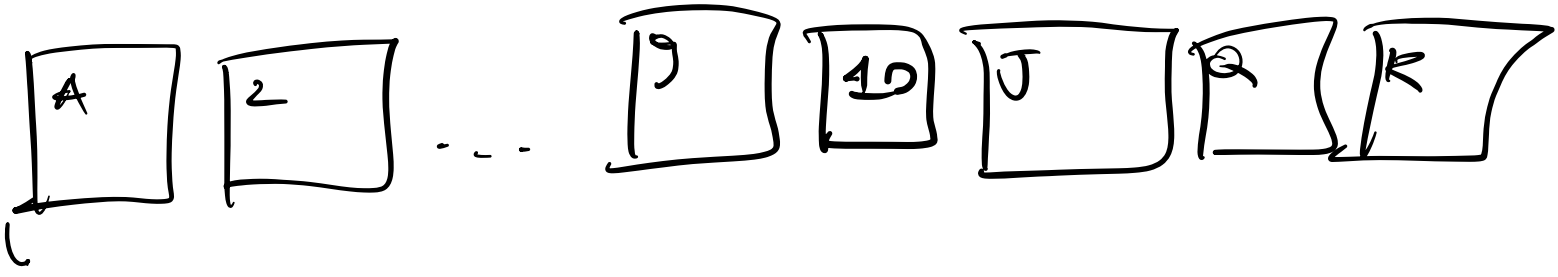
Un'

4

CARTE DA POKER

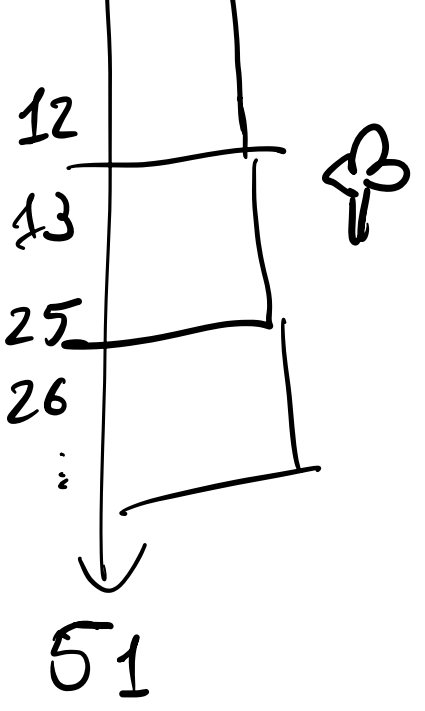
SEMI





13

$4 \times 13 = 52$



xfr2 o e 51

func nomeCarta(x int) string {

var seme, val string

switch x/13 {

case 0: seme = "♣"

case 1: seme = "♠"

case 2: seme = "♦"

case 3: seme = "♥"

}

switch x%13 {

case 0: val = "2"

```
case 1 :
    :
case 7 :
case 8 :
case 9 :
case 10 :
case 11 :
case 12 :
}
return val + serve
}
```

val = "3"

val = "9"

val = "10"

val = "J"

val = "Q"

val = "K"

val = "A"

(*)

```
if x % 13 <= 8 {
    val = strconv.Itoa((x % 13) + 2)
} else {
```

switch x%13 {

case 9:
case 10:
case 11:
case 12:

}

val = "J"
val = "Q"
val = "K"
val = "A"

(8)

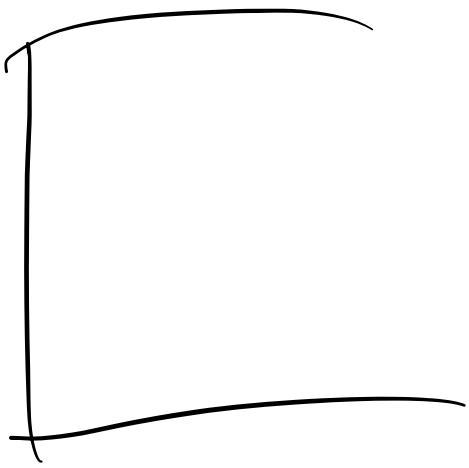
}

string (105) → "H"

strconv.Itoa (105) → "105"

FUNZIONI CHE RESTITUISCONO PIÙ VALORI

```
func Pippo(...) int, string {
```



```
}
```

COMMA-OK
COMMA-ERROR
} IDIOM

true: tutto bene OK
false: qualcosa è andato male

```
func pippo(...) int, bool {
```

```
}
```

```
x, ok := pippo(...)
if ok {
    x....
} else {
    fmt.Println("Problem nella
    chiamata alla funzione
    pippo")
}
```

```
func pluto(...) int, error
                    ↓
                   nil
```

```
}
x, err := pluto( ... )
if err == nil {
        ... x ...
} else {
        c'è stato un errore
}
```

ESEMPIO

- Numero misterioso: dsfd
una stringa

s *1**2* 37***

s → numero

*1**2* → 1237

```
func numMisterioso(s string) int {
    pulita := ""
    for _, r := range s {
        if r != '*' {
            pulita += string(r)
        }
    }
    num, err := strconv.Atoi(pulita)
    if err == nil {
        return num, true
```

```
            return n, true
        } else {
            return 0, false
        }
    }
x, ok = numMisterioso("**72*34")
```