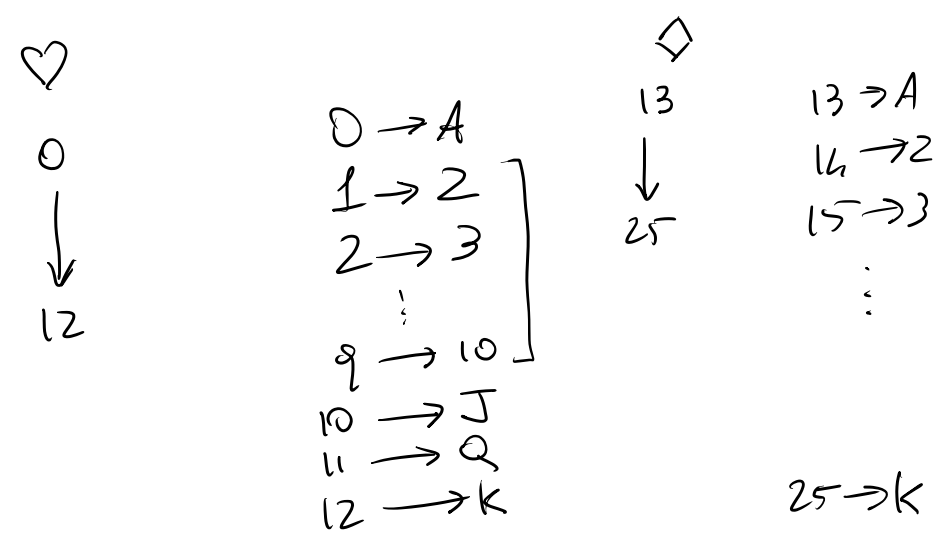
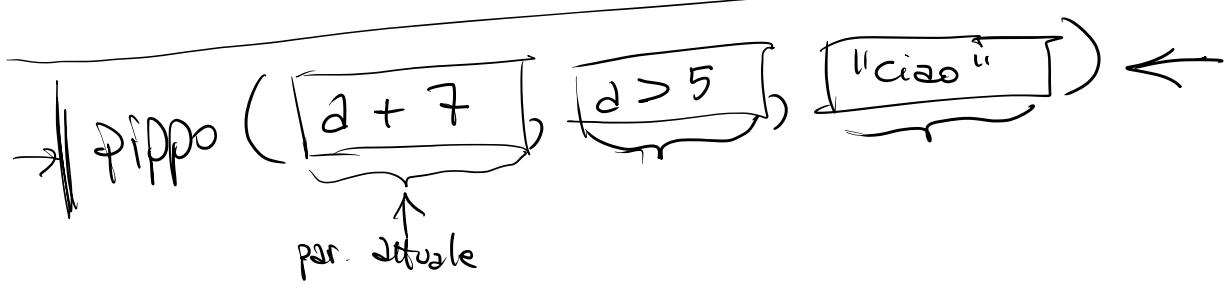


```

func pippo ( par. formale x int, y bool, z string ) {
  if {
    return
  }
}

```



func conv (x int) string }

var res string

switch x%13 {

case 0: res = "A"

case 10: res = "J"

case 11: res = "Q"

case 12: res = "k"

default : res = strconv. Itoa (x%13+1)

~~y := x%13+1
res = string(y)~~

}
switch x/13 {
case 0: return rest "0"
case 1: return rest "1"
case 2: return rest "2"
default : return rest "3"
}

}

func cont (x int) (res string) {

```

switch     x%13 {
case     0:     res = "A"
case     10:    res = "J"
case     11:    res = "Q"
case     12:    res = "K"
default :     res = streamv. Itoa (x%13 + 1)

```

```

}
switch     x/13 {
case     0:     rest = "m"
case     1:     rest = "n"
case     2:     rest = "o"
default :     rest = "p"
}

```

return

}

* Data una stringa x restituisce, nell'ordine,
 il numero di vocali latine minuscole
 non accetate e il numero di
 consonanti. func (x string) * (int, int) }

```

var v, c int
for _, r := range x {
  switch r {
    case 'a', 'e', 'i', 'o', 'u': v++
    case 'b', 'c', 'd', ..., 'z': c++
  }
}
return v, c
}

```

```

func main() {
  var s string
  fmt.Scan(&s)
  a, b := contVoe Cons(s)
  fmt.Println("Ci sono", a, "vocali e", b,
             "consonanti")
}

```

func countVocCons (x string) (v int, c int) {

for _, r := range x {

switch r {






case 'a', 'e', 'i', 'o', 'u': v++

case 'b', 'c', 'd', ..., 'z': c++

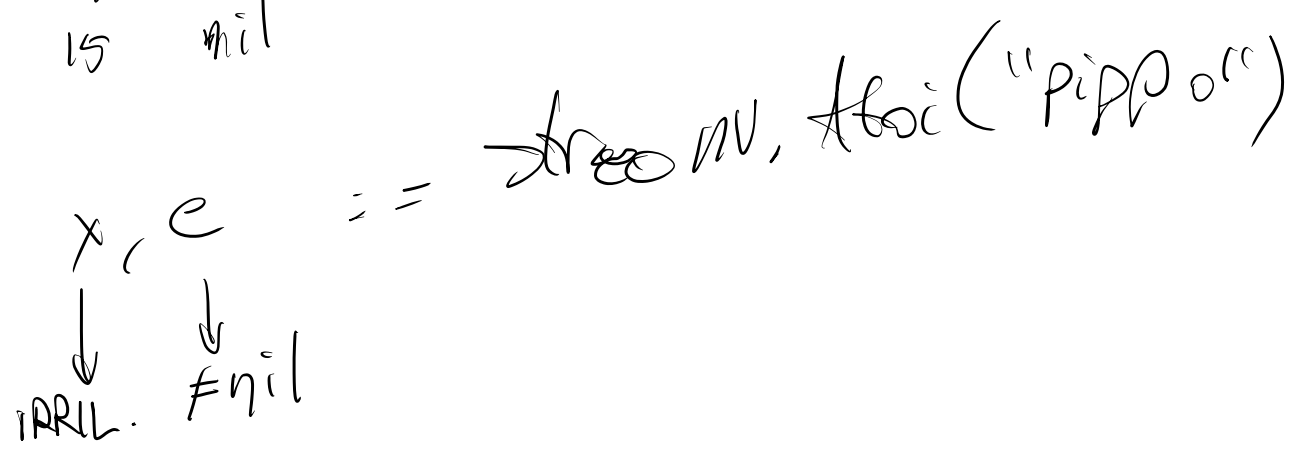
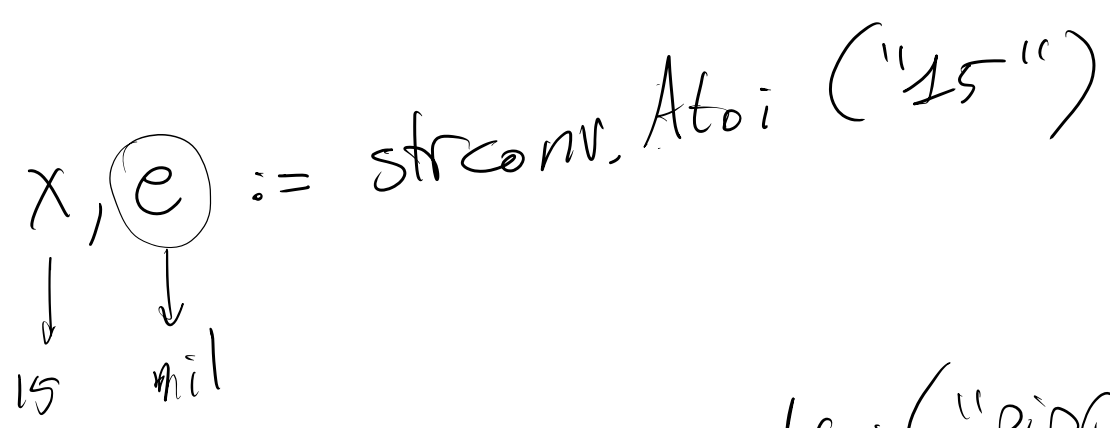
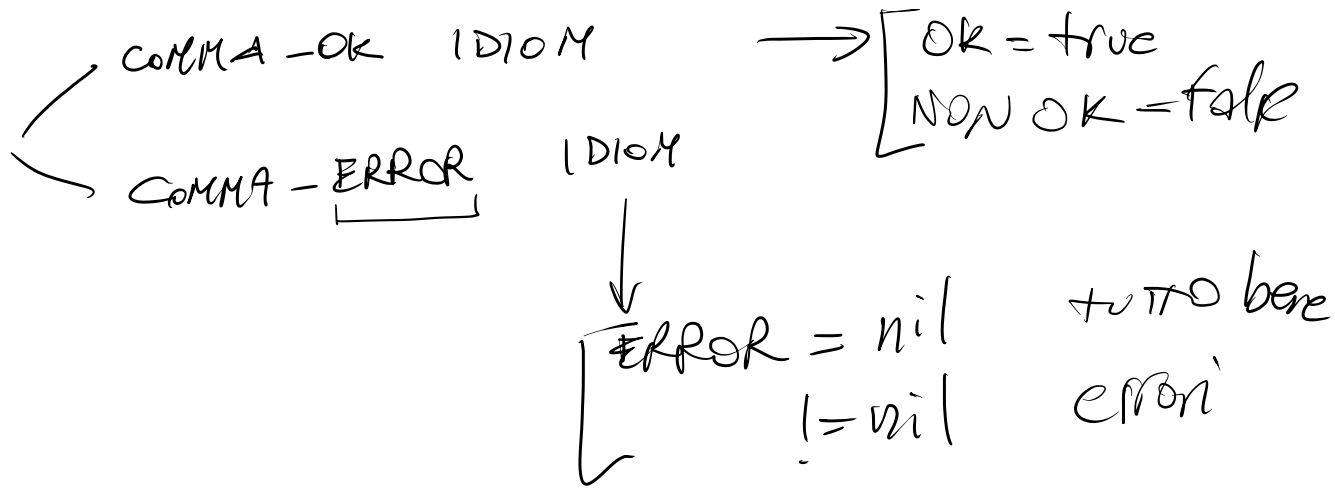
}

}
return

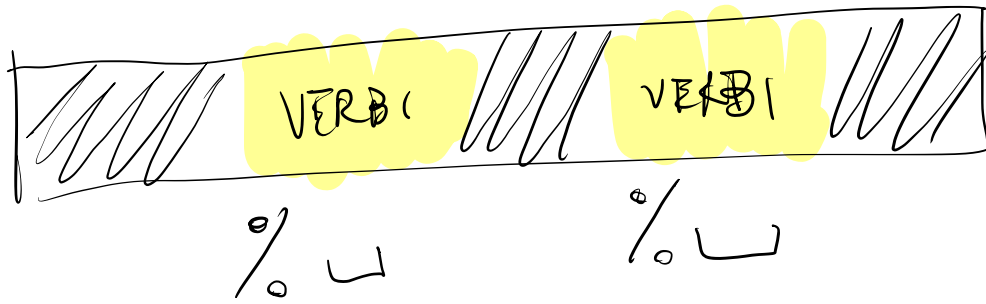
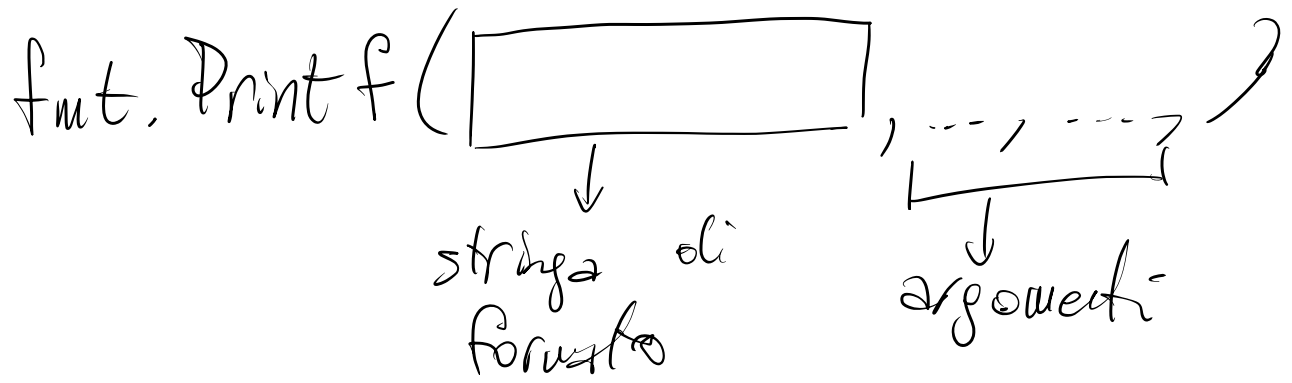
}

switch {
case , ,  :
[]
case ,  :
[]
}

switch {
case $x > y$, ~~$x < 1$~~ , ...:
[]
case $x * x == 5$:
[]



data una stringa 2 che corrisponde
2 una data ("7/11/2022")
restituire giorno, mese, anno.



%d	intero
%f	floating-point
%s	stringa

```

func main() {
  var s string
  funt. Scan (&s)
  a, b := contorbe Cons (s)
  funt. Println ("Ci sono", a, "vocali e", b,
                "consonanti")
}

```

```

funt. Printf ("Ci sono %d vocali e %d
             consonanti\n", a, b)

```

AVVERBI

- %f
- %.2f
- %7.2f

$x := \text{fmt.Sprintf}(\text{"ci sono } \%d \text{ vocali"} \\ v)$

- SCRIVERE UNA FUNZIONE CHE
DATA UNA DATA g, m, d
(toe int) LA RESTITUISCE
COME STRINGA (es. "7/14/2001")