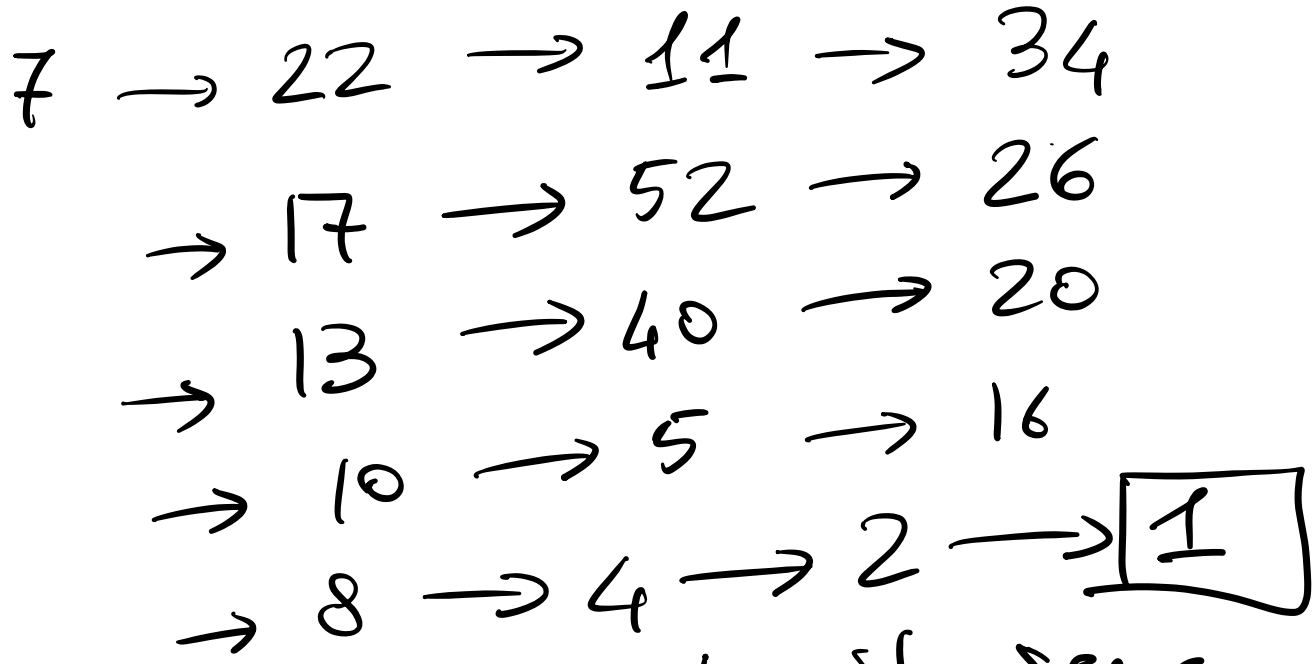
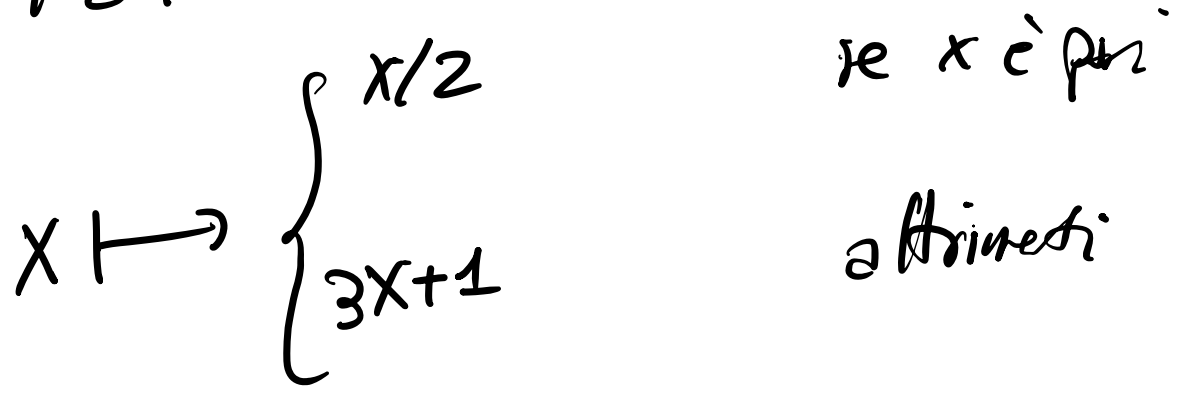


ESERCIZI

- SEQUENZA DI COLLAZIONE



Dato in input il numero
stampare la sequenza

— Epoch

1/1/1970

Scriviamo una funzione
che dato una data (g, m, a)
restituisca i giorni passati
da quella data.
dall'epoca

```
func daysFromEpoch (g, m, a int) int {  
  var count int  
  for x := 1970; x < a; x++ {  
    count += 365  
    if isLeap(x) {  
      count++  
    }  
  }  
  for x := 1; x < m; x++ {  
    if x == 1 || x == 4 || x == 6 ||  
      x == 9 {  
      count += 30  
    } else if x == 2 {  
      count += 31  
    }  
  }  
}
```

MESI

if isleap(a) {
count += 29

} else {
count += 28

} else {
count += 31

}

}

count += u

return count

}

func

isleap (a int) bool {
return (a%4 == 0 && a%100 != 0)
|| a%400 == 0

}

29/11/2023

data: ...
Sono passati ... di giorni

func parseDate (s string) (int, int, int)
b1 := strings.Index(s, "/")
b2 := strings.Index(s[b1+1:], "/")
 + b1 + 1
s_g := s[:b1]
s_m := s[b1+1:b2]
s_a := s[b2+1:]
g, - := strconv.Atoi(s_g)
m, - := strconv.Atoi(s_m)
a, - := strconv.Atoi(s_a)
-
return g, m, a

```

}
func main() {
  var s string
  var g, m, a int

  fmt.Scan(&s)
  g, m, a = parseDate(s)
  c := daysFromEpoch(g, m, a)
  fmt.Println("Sono passati", c,
    "giorni dall'epoca")
}

```

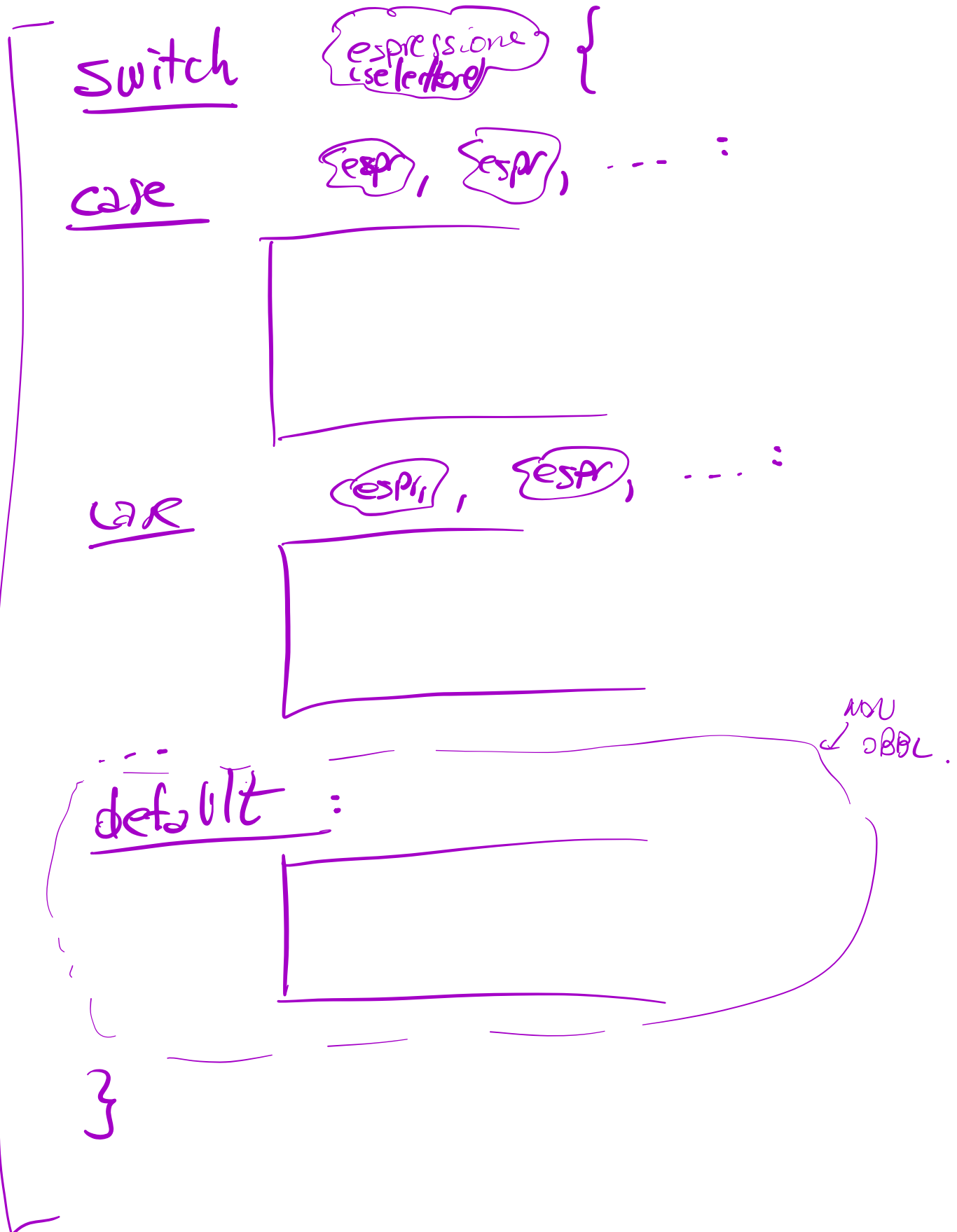
Esercizio

- rendere più robusto il parseDate usando il COMMA/OK idiom

COMMA / OK IDIOM
COMMA / ERROR IDIOM

"1231" → 1231

SELEZIONE MULTIARIA (SWITCH)



```
if x == 11 || x == 4 || x == 6 ||  
x == 9 {
```

```
    count += 30
```

```
} else if x == 2 {
```

```
    if isLeap(a) {  
        count += 29
```

```
    } else {
```

```
        count += 28
```

```
} else {
```

```
    count += 31
```

```
}
```

```
switch (x) {
```

```
    case "1, 4, 6, 9":  
        count += 30
```

```
    case 2:
```

```
        if isLeap(a) {  
            count += 28
```

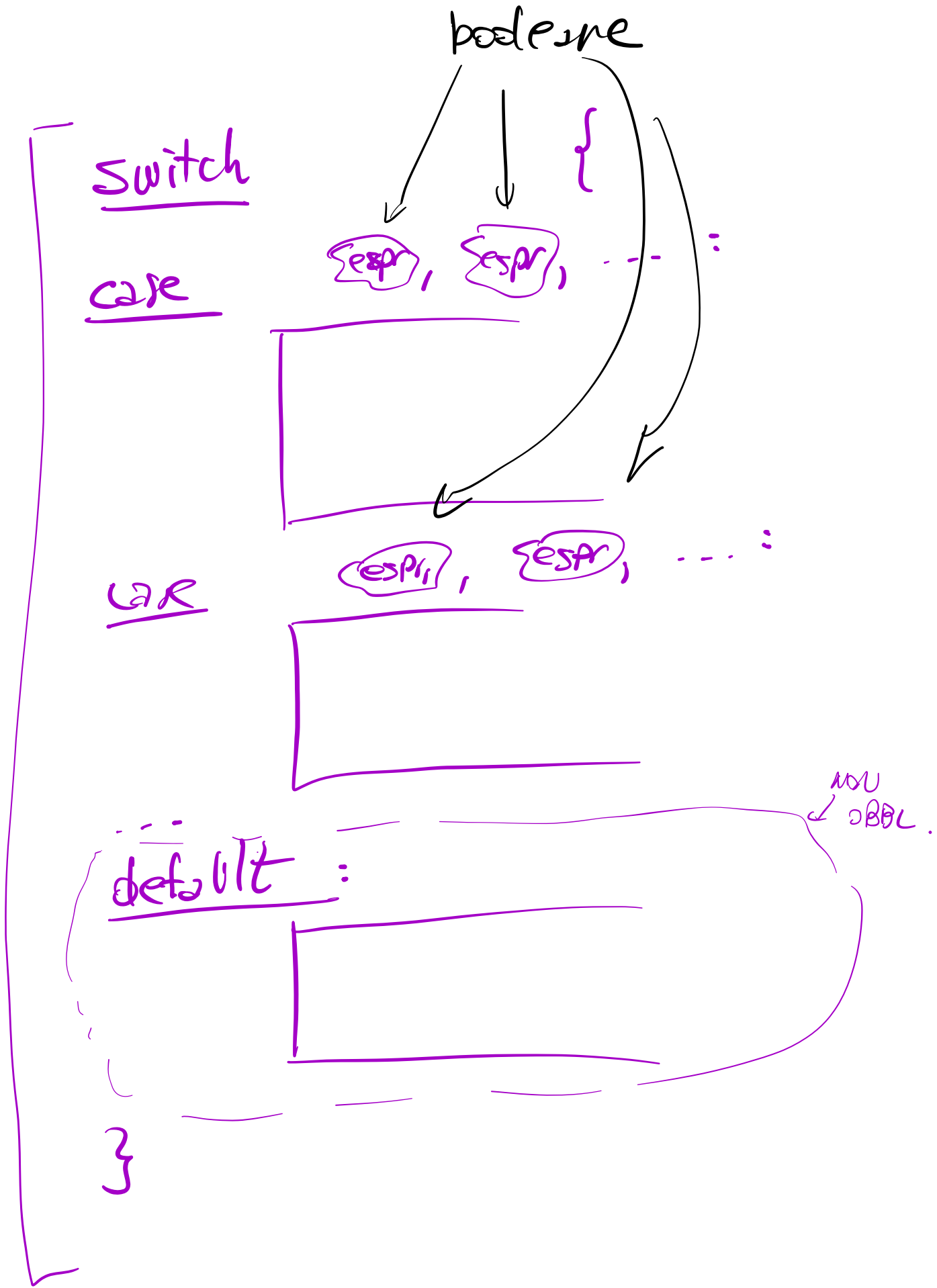
```
        } else {  
            count += 29
```

```
        }
```


default : count += 31

}

USO SPECIALE DI SWITCH



Case fare

switch true {

}

switch {

case $x > 0$ $y < 0$:

A

case $x == 0$ && $y > 0$:

B

default :

P

?

POKER

4
- ♡, ♦, ♣, ♠

- A, 2, ..., 10, J, Q, K
13

52

0, ..., 12: carte di ♡
13, ..., 25: " " ♦
⋮

Scrivere una funzione
che dato il numero di
una carta (0 → 51)
restituisca una stringa

con la carta
("10 □")