

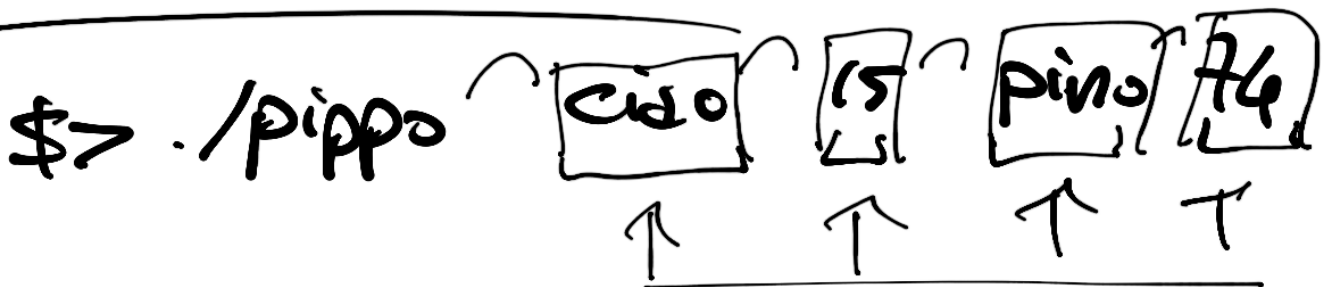
ARGOMENTI SULLA RIGA DI COMANDO

\$> ./pippo



\$>

\$> .\pippo.exe



os. Args

[] string
0 → nome del programma
1 }
2 } → argomenti sulla
 } riga di comando
 :
 n

calc.go

\$> ./calc 15.7 + 3
18.7
\$>

15.7 + 3
1
2 3

package main
import (
 "fmt"
 "os"

)
func

if main() {
 len(os.Args) != 4 {
 fmt.Printf("Ci sono %d "+
 "argomenti!\n", len(os.Args))
 return
 }
}

```
a, err := strconv.ParseFloat(os.Args[1], 64)
if err != nil {
    fmt.Println("%s non è un numero",
        os.Args[1])
    return
}
b, err := strconv.ParseFloat(os.Args[2],
```

```
64)
if err != nil
```

```
var risultato float64
switch os.Args[2] {
```

```
case "+":
    risultato = a + b
```

```
case "-":
    risultato = a - b
```

```
default:
```

```
    fmt.Printf("%s non è
un operatore consentito", os.Args[2])
    return
}
```

```
fmt.Printf("%f %s %f = %f\n",  
2, os.Args[2], b, risultato)
```

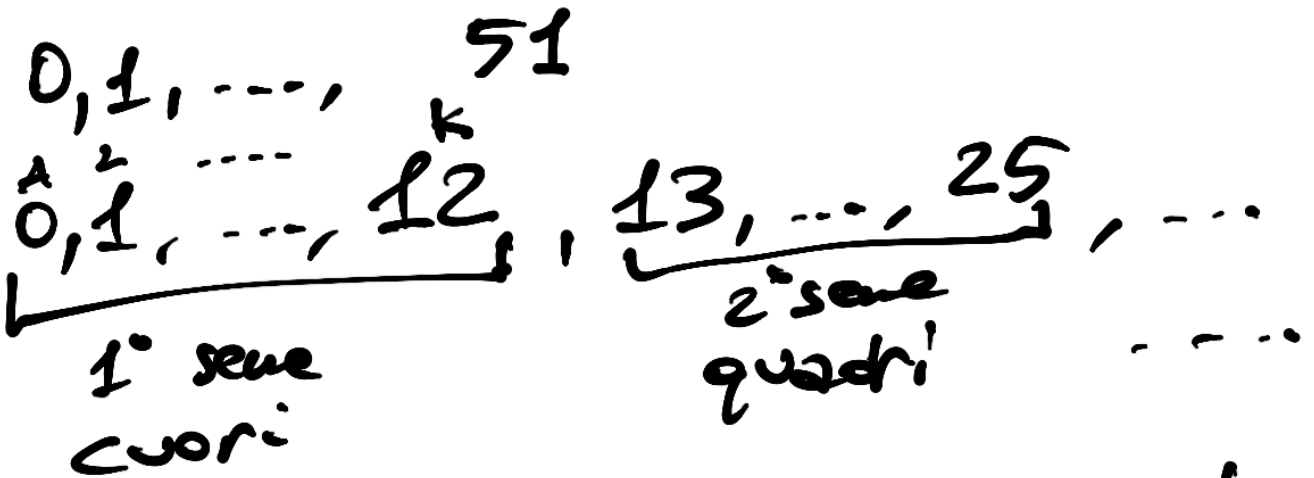
```
}
```

```
$> ./calc 13.7 - 2.4  
13.7 - 2.4 = 11.3  
$>
```

```
$> ./calc 3.7 x 2  
./calc 3.7 calc.go pippo.txt... 2
```

POKER

$$4 \text{ semi} \times 13 \text{ valori} = 52 \text{ carte}$$



```
func seme(c carta) string {  
    ...  
}  
func valore(c carta) string {  
    ...  
}
```

```
func String(c carta) string {  
    valore(carta) + seme(carta)  
}
```

K♥
10♠

```
type (  
    carta int  
    mazzo [ ] carta
```

```
)  
func catoleria() (ris mazzo) {
```

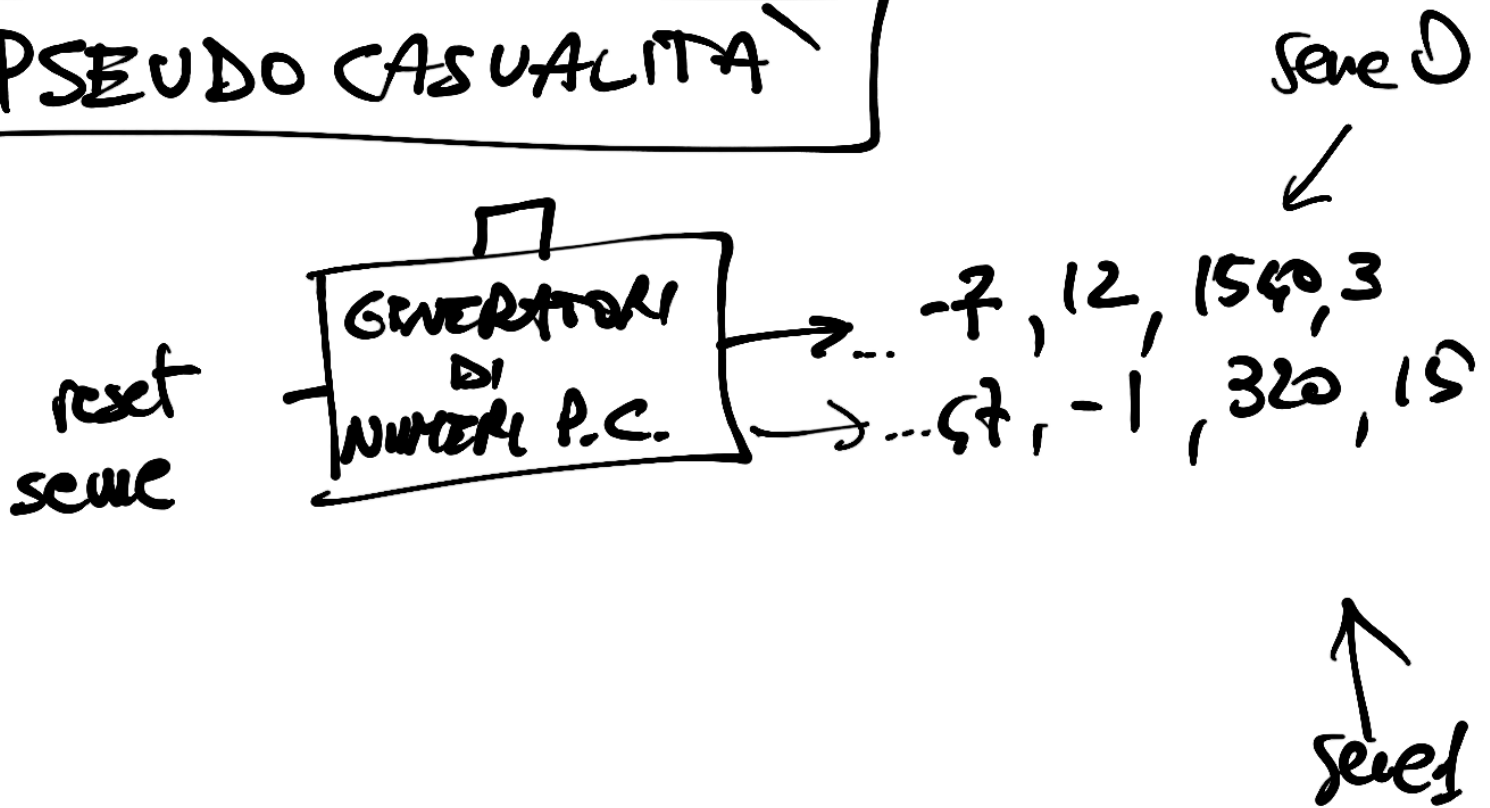
```
    for i:=0; i<52; i++ {  
        ris = append(ris, carta(i))
```

```
    }  
    return
```

```
}
```

```
func stampa Mazzo (n uazzo) {  
  for _, c := range n {  
    fmt.Println (String(c))  
  }  
}
```

PSEUDO CASUALITA'



$$\underline{X_{n+1}} = \left(\underset{\substack{\uparrow \\ 3}}{2} X_n + \underset{\substack{\uparrow \\ 6}}{C} \right) \bmod \underset{\substack{\uparrow \\ 5}}{m}$$

$$\underline{X_{n+1}} = \left(\underset{\substack{\uparrow \\ \textcircled{3}}}{2} \underline{X_n} + \underset{\substack{\uparrow \\ \textcircled{6}}}{c} \right) \underline{\text{mod}} \underset{\substack{\uparrow \\ \textcircled{5}}}{m}$$

X_n	X_{n+1}
1	2
4	2
3	⋮
0	⋮
1	
⋮	

math/random

- randInt()
- rand.Seed(15)

time.Now().UTC().UnixNano

func main() {
rand.Seed(time.Now().UTC().UnixNano)
:
}

COME SI MESCOLA UN
MAZZO?

```
func mazziera() (ris mazzo) {  
  for i:=0; i<52; i++ {  
    ris=append(ris, carta(i))  
  }  
}
```

carta(rand.Int() % 52)



```

func presente (m wazto, c cartz) bool {
  for -, x := range m {
    if x == c {
      return true
    }
  }
  return false
}

```

```

func waziere () (res wazto) {
  for len(res) < 52 {
    c := cartz(rand.nextInt() % 52)
    if ! presente(res, c) {
      res = append(res, c)
    }
  }
}

```

```

  return

```

```

func mescol (m array) {
  for i:=0; i<52; i++ {
    j = i + rand.Int()%(52-i)
    m[i], m[j] = m[j], m[i]
  }
}

```

↳

COME SI GENERA UN ITERO
 A CASO DA 0 COMP. A M ESCI?

rand. Int() % n

A CASO DA A COMP. A B ESCI.

A + rand. Int() % (B-A)

func main () {
 rand.Seed (- - - -)
 m := cartoleria ()
 mescola (m)
 stampa Mazzo (m)
 mano := n [0: 5]
 stampa Mazzo (mano)

```

func   isCoppia (mano mano) bool int
  for   i := 0; i < len(mano); i++ {
    for   j := i+1; j < len(mano); j++ {
      if   valore(mano[i]) ==
            valore(mano[j]) {
                return true
            }
    }
  }
  return false, -1
}

```

i	j
0	1, 2, 3, 4
1	2, 3, 4
2	3, 4
3	4

/* Data una vanao, guarda tutte le
coppie e le restituisce sottofans
di slice dei loro valori.
Assume che la vanao non contenga
né tris né poter */

```
func coppiaPotente(vanao vanao)
    (res []int) {
    for i := 0; i < len(vanao); i++ {
        for j := i+1; j < len(vanao); j++ {
            if vanao[i] ==
                vanao[j] {
                res = append(res,
                    vanao[i])
            }
        }
    }
    return res
}
}
```