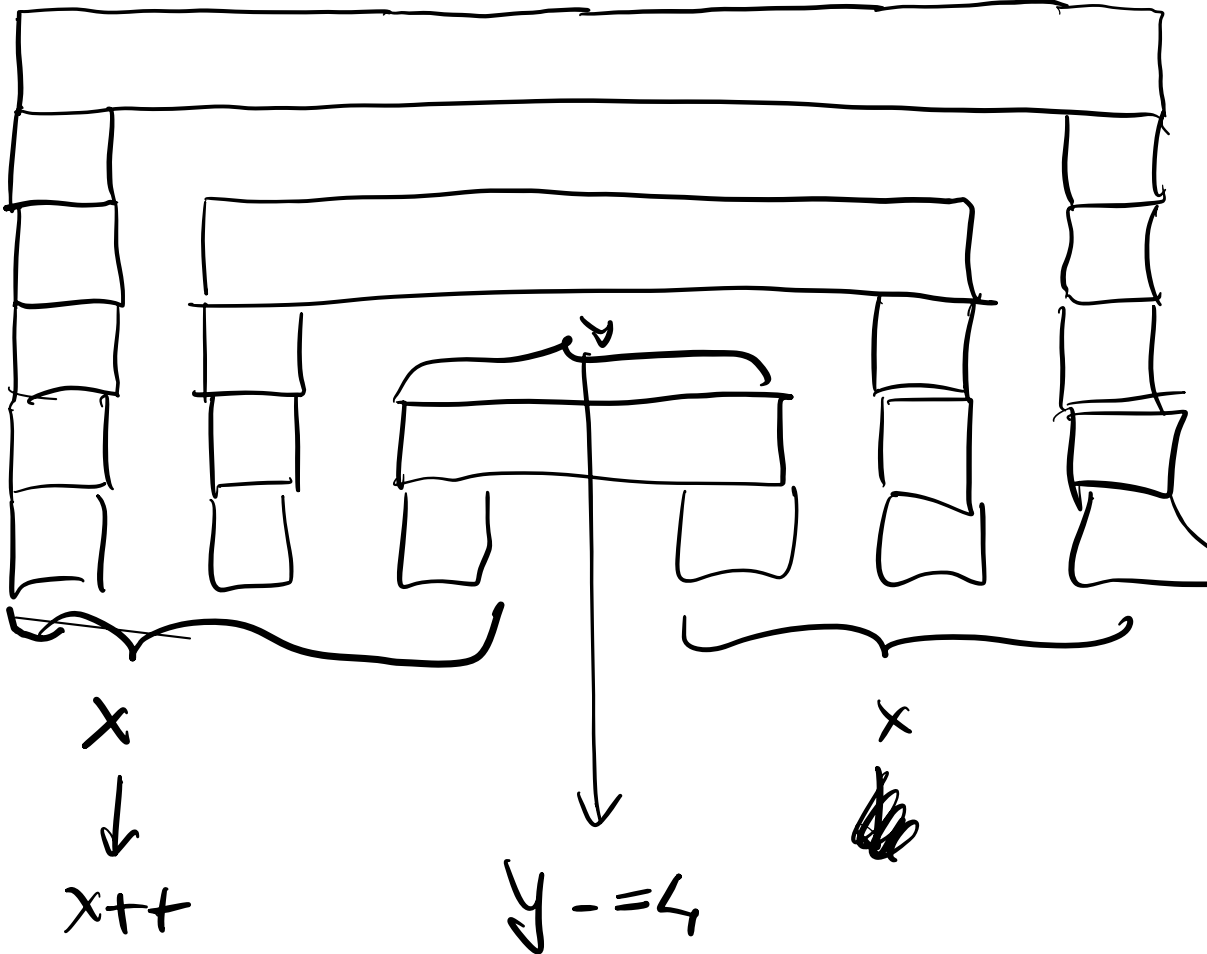
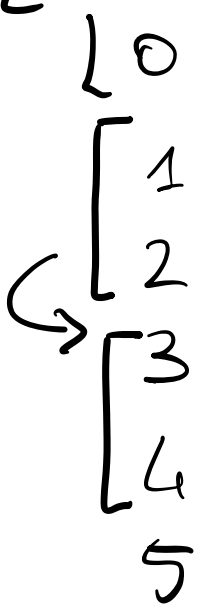


M=12



x:=0
y:=n
for

```

i:=0; i < M/2; i++ }
  for j:=0; j < x; j++ }
    fmt.Print("* ")
  }
  for j:=0; j < y; j++ {
    if i%2 == 0 {
      fmt.Print("*")
    } else {

```

```
        fact. Print("L")
    }
}
```

```
for j:=0; j<x; j++ {
    fact. Print("L*")
}
```

```
fact. Println()
```

```
if i%2 == 0 {
    if i < n/2 {
        x++
    } else {
        y-- = 4
        x--
        y++ = 4
    }
}
```

}

type

Carta struct {

sewe int

// 0 = wari, 1 = quadri, ...

vibre int

// 0 = A, 1 = 2, ..., 9 = 10,
10 = J, 11 = Q, 12 = K

}

func

mazzo() [] Carta {

var m [] Carta

for s := 0; s < 4; s++ {

for v := 0; v < 13; v++ {

var c Carta

c.sewe = s

c.vibre = v

m = append(m, c)

} }

return m

}

func mescola (m []Carta, r *rand.Rand)

// Con rejection

m := len(m)

res := make([]Carta, m)

usato := make([]bool, m)

for i := 0; i < m; i++ {

for {

j := r.Intn(m)

if !usato[j] {

break

} }

res[i] = m[j]

usato[j] = true

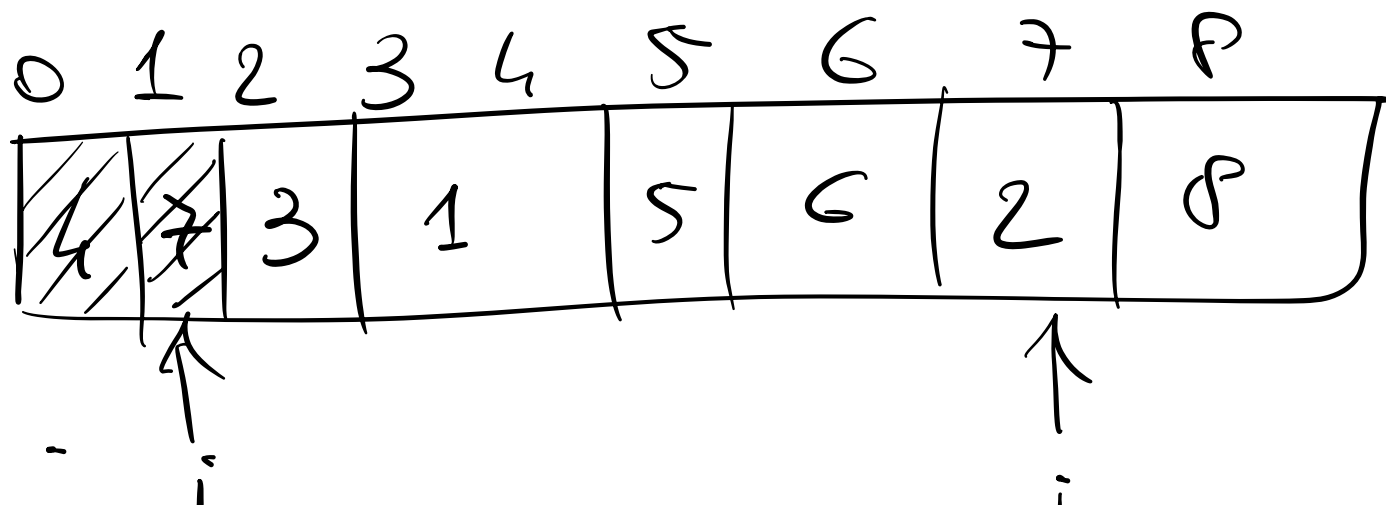
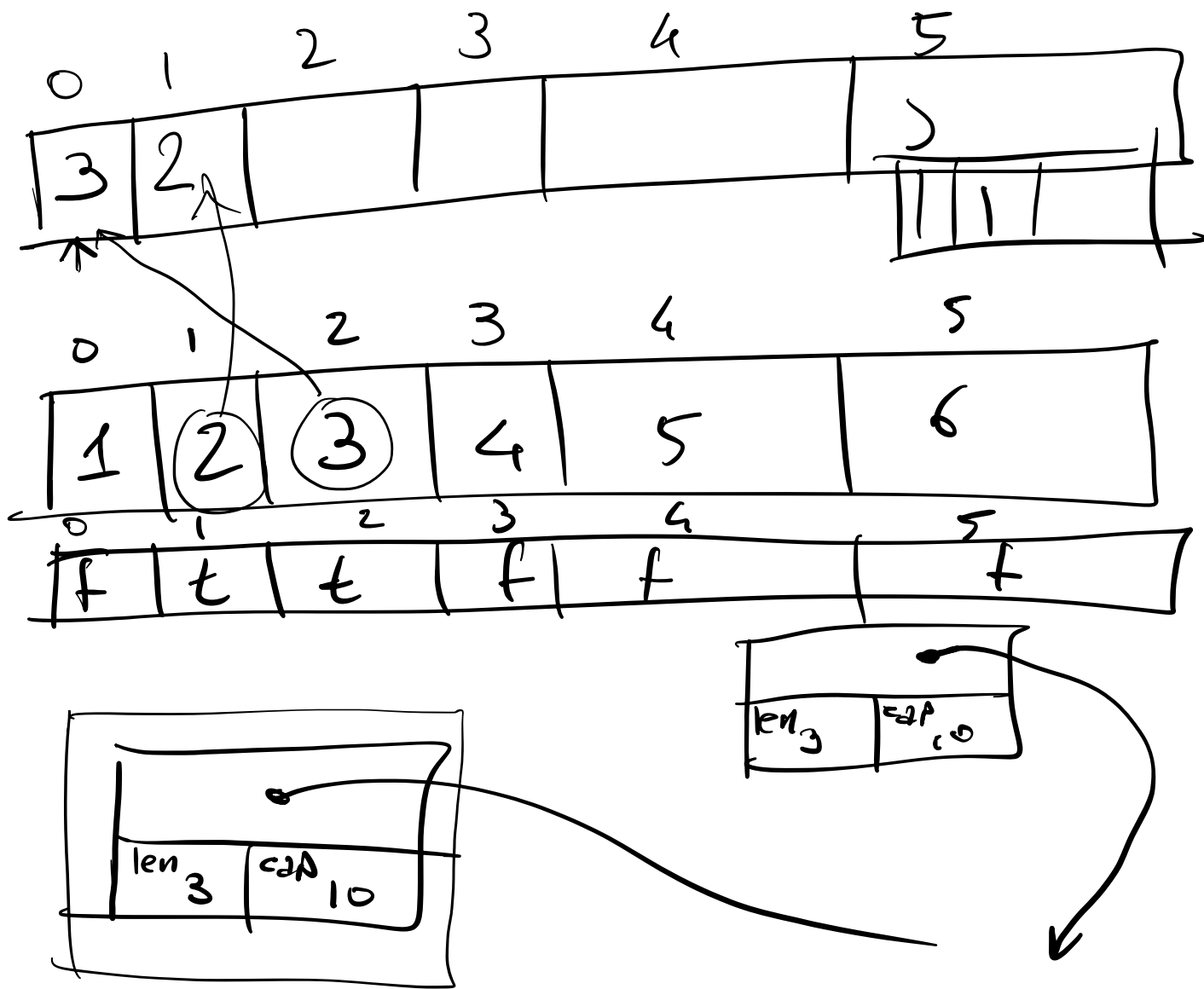
}

for i := 0; i < m; i++ {

m[i] = res[i]

}

}



```
func mescola(m []Carta, r *rand.Rand)
```

```
  n := len(m)  
  for i := 0; i < n; i++ {
```

```
    j := i + r.Intn(n-i)
```

$0, 1, \dots, n-i-1$
 $i, i+1, \dots, n-1$

```
    m[i], m[j] = m[j], m[i]
```

```
  }
```

```
  }
```

```
func stampaCarte(m []Carta) {
```

```
  for _, c := range m {  
    fmt.Println(strCarta(c))
```

```
  }
```

```
}
```

func

```
main () {  
  r := rand.New(rand.NewSource(  
    time.Now().UnixNano()))
```

```
  m := m2z(r)
```

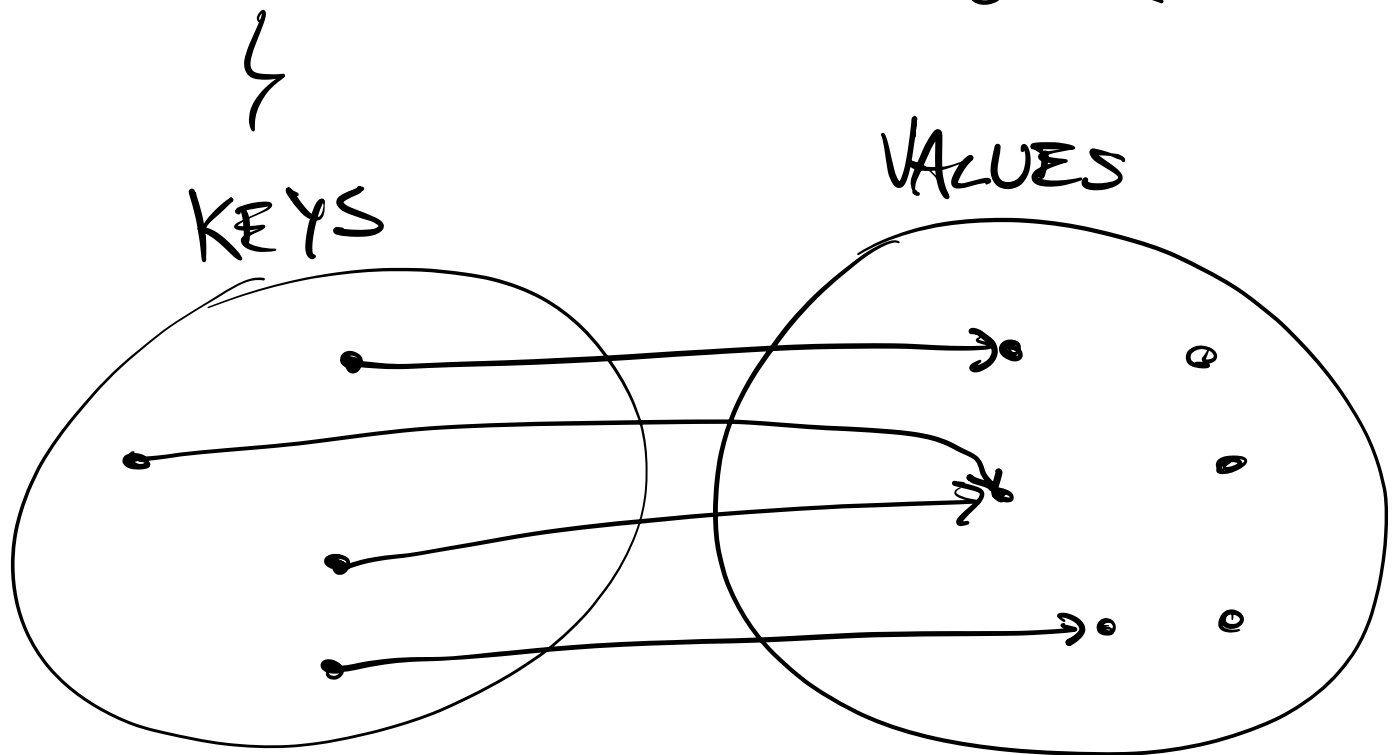
```
  mescola(m, r)
```

```
  stampaCarte(m[:5])
```

```
}
```

MAPPE

```
type Personz struct {  
  nome, cognome string  
  nazionalità dato  
  cf string  
}
```



map [tipo chiave] tipo valore

↓
make

DICHIARAZIONE

var d map[string]int

CREAZIONE

d = make(map[string]int)

POPOLAZIONE

d["Paolo Boldi"] = 172

d["Anna Bolena"] = 164

d["Giovanni Rubini"] = 184

d["Giulio Cani"] = 172

ACCESSO

fmt.Println(d["Paolo Boldi"])

fmt.Println(d["Pippo"])

↙
○

ACCESSO
CENTRALIZZATO

v, ok := d["Pippo"]

CANCELLAZIONE
CHIAVE

delete(m, "Giulio Cani")

{ for k, v := range m }

ITERA
LE COPPIE
CHIAVE/
VALORE

for $\{$

$\{$

$\}$
 $\}$
 $\}$

ESERCIZIO

Scrivere una funzione che
data una stringa trovi quante
volte ogni carattere compare

```
func freq (s string) map[rune]int {  
    var res map[rune]int  
    res = make(map[rune]int)  
    for _, r := range s {  
        res[r]++  
    }  
}
```