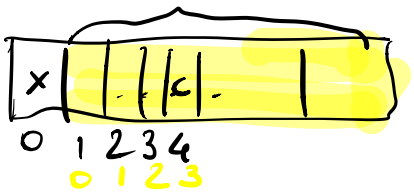
Scrivere una funzione ricorsiva che data una stringa ASCII s e un byte (= carattere) c, trovi e restituisca la prima pos. di c in s

VERSIONE ITERATIVA

```
func pos (s string, c byte) int {
    for i:=0; i < len(s); i++ {
        if s[i] == c {
            return i
        }
    }
    return -1
}
```

s

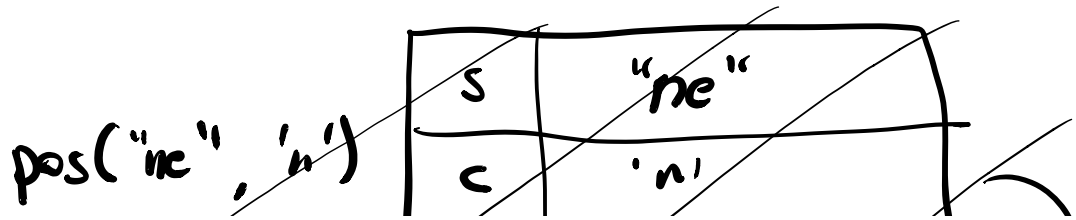


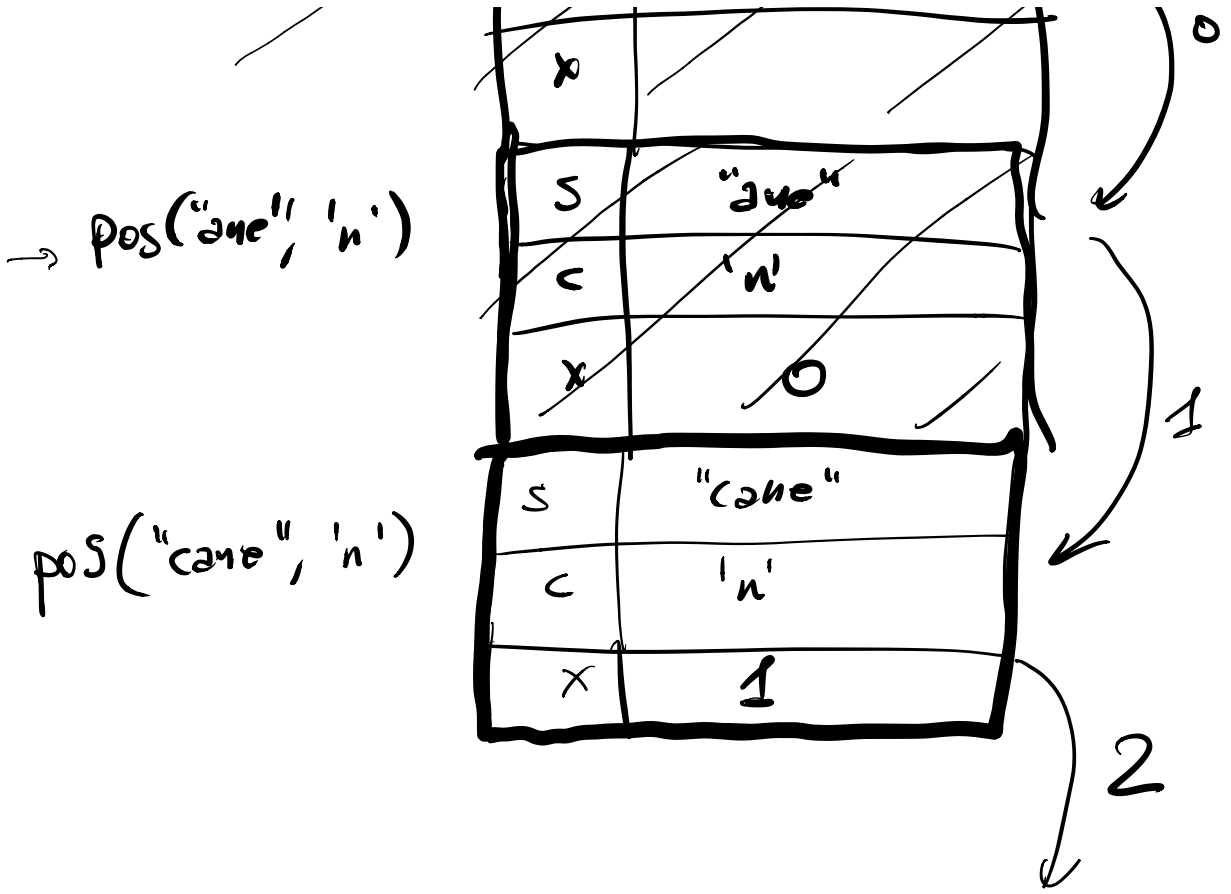
VERSIONE RICORSIVA

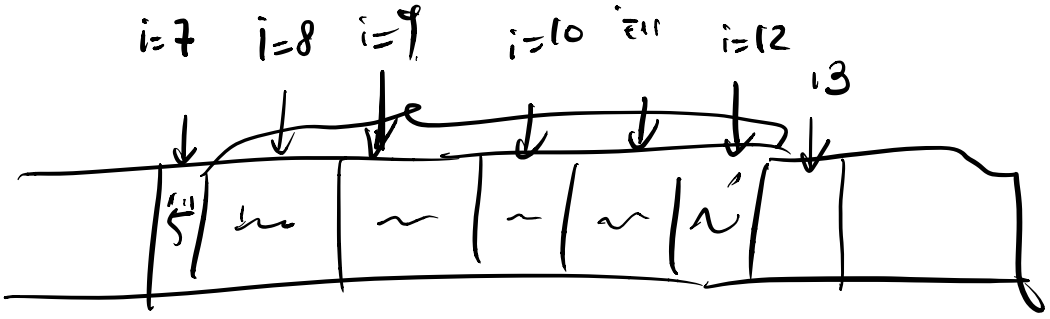
```
func pos (s string, c byte) int {
    if len(s) == 0 {
        return -1
    }
    if s[0] == c {
        return 0
    }
    x := pos(s[1:], c)
    if x == -1 {
        return -1
    }
    return x + 1
}
```

CASO BASE

RICORS

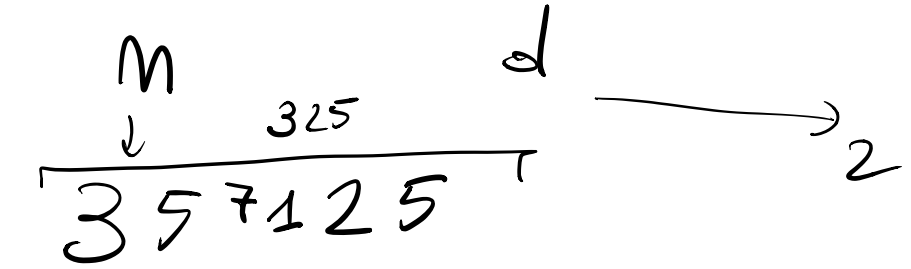



x
s
"gue"
c
'n'
x
0
s
"cane"
c
'n'
x
1
0
1
2
→ pos("gue", 'n')
pos("cane", 'n')


i=7
i=8
i=9
i=10
i=11
i=12
13

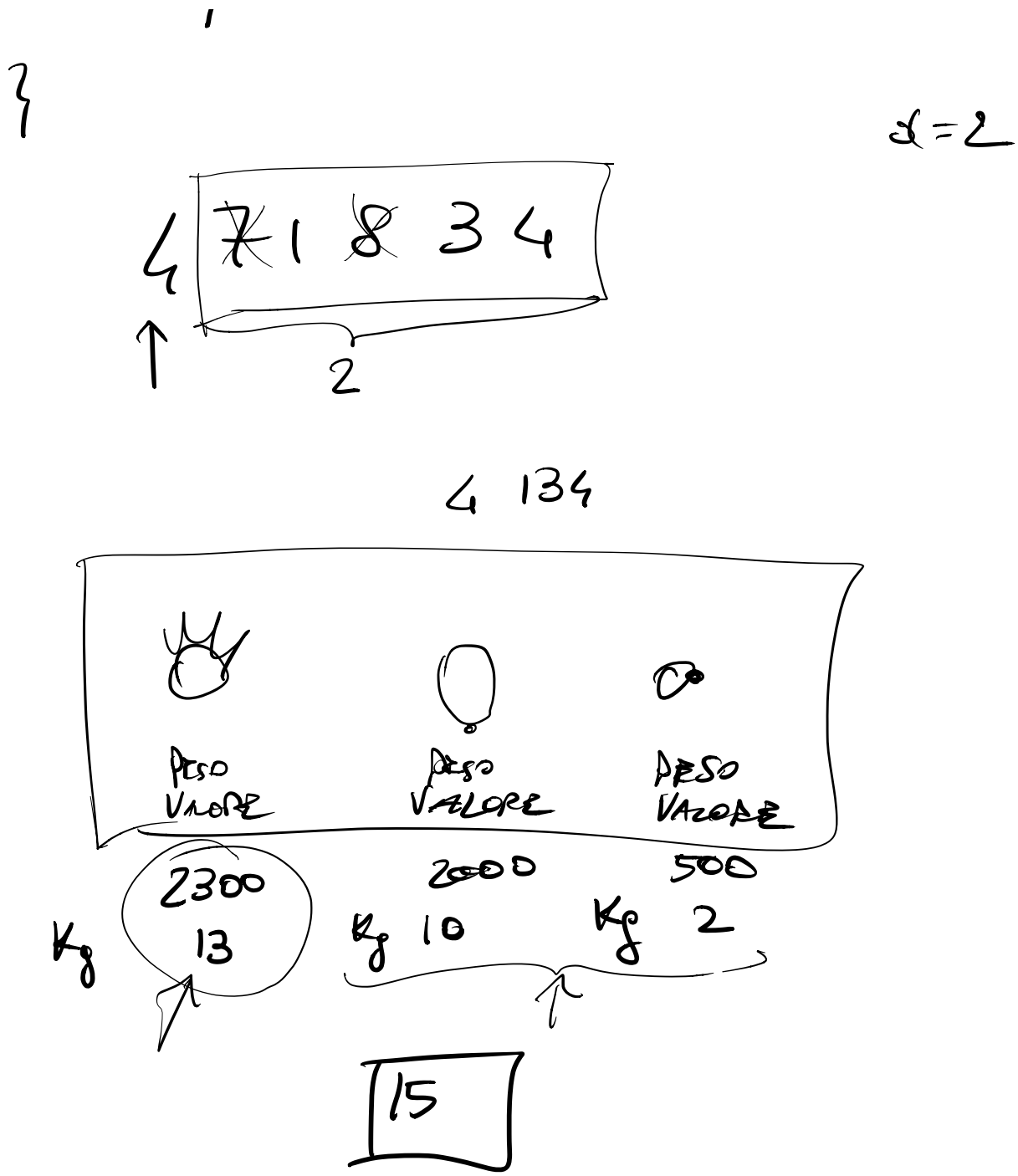
```
func cancelb (s []string) []string {
    var res []string
    for i:=0; i< len(s); i++ {
        n, ok := strconv.Atoi(s[i])
        if !ok {
            res = append(res, s[i])
        } else {
            i += n
        }
    }
    return res
}
```

a = append(a, b...)



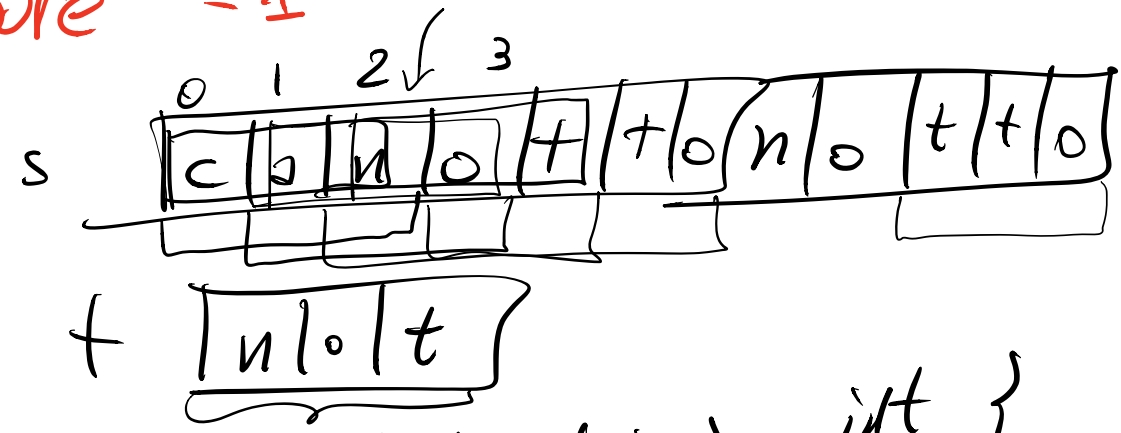

```
func f(n string, d int) int {
    if d == 0 {
        x, _ := strconv.Atoi(n)
        return x
    }
    m1 := f(n[1:], d-1)
    y := f(n[1:], d)
    m2, _ := strconv.Atoi
        (string(n[0]) +
        strconv.Itoa(y))
    if m1 < m2 {
        return m1
    } else {
        return m2
    }
}
```

"3_m2_1"


1
}
d = 2
4 7 1 8 3 4
2
4 134
PESO
VALORE
PESO
VALORE
PESO
VALORE
2300
2000
500
Kg 13
Kg 10
Kg 2
15

Scrivere una funzione che data una stringa s e un'altra stringa t, trova la posizione in s in cui t compare per la prima volta come sottostringa oppure -1

[Ascii]



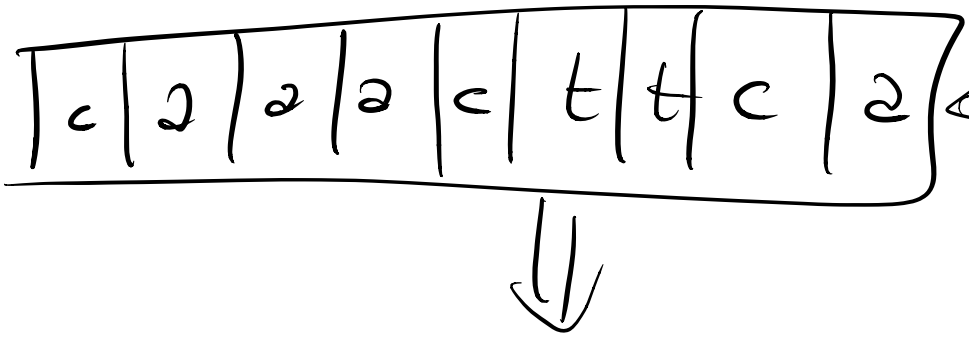

```
func search(s, t string) int {
    nt := len(t)
    for i := 0; i < len(s)-nt; i++ {
        if s[i : i+nt] == t {
            return i
        }
    }
    return -1
}
```

[NON ASCII]

```
func search(s, t string) int {
    c := 0
    nt := len(t)
    for i, _ := range s {
        if i+nt < len(s) && s[i:i+nt] == t {
            return c
        }
        c++
    }
    return -1
}
```

s "Una casa troppo piccola"

a



"Una 3454 troppo ..."

func transform(s string, a []rune) string {

(X)

```
    var m map[rune]int
    m = make(map[rune]int)
    for _, r := range a {
        m[r]++
    }
    res := ""
    for _, r := range s {
        if m[r] == 0 {
            res += string(r)
        } else {
            res += strconv.Itoa(
                m[r])
```