STRING & RUNE CHEAT SHEET

RUNE

- singoli caratteri
- rune = int32

```
var c,d rune
if c>='a' && c<='z' {

}
d = rune(c-'a'+'A')
x:= c-'0'
```

unicode

STRING

- sequenze di caratteri

""

-

"a"   'a'

$\underbrace{\text{"pippo"}}$

```
var s string
s = "pippòlo"
```

```
len(s)
s[i]
```

```
for i,c := range s {
```

}

CONCATENAZIONE

$$\underbrace{\text{stringhe} + \text{stringhe}}_{}$$

```
var   x, y   string
  x = "pippo"
  y =  x  +  " è tornato"
  x = "Garibaldi"
```

y = x + " è tornato"

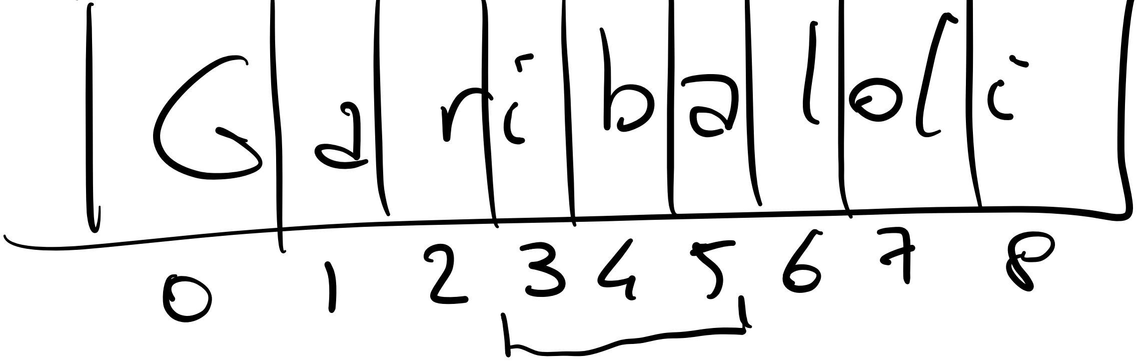
SUBSTRING

stringa [i : j]

indice di partenza (COMPRESO)

indice di arrivo (escluso)

```
var x, y string
x = "Garibaldi"
y = x[3:6]  ← "iba"
```



G	a	r	i	b	a	l	d	i
0	1	2	3	4	5	6	7	8

$$y = \underbrace{x + x[:3] + x}_{\text{"GaribaldiGarGaribaldi}}$$

INPUT DI STRINGHE

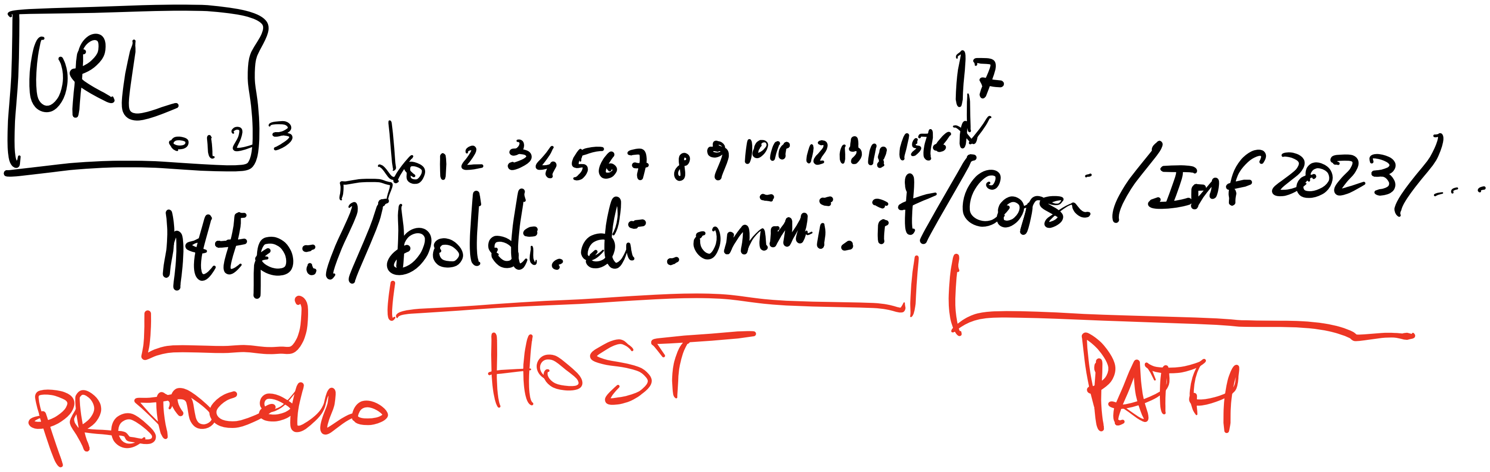
```
var s string
fmt.Scan(&s)
```

DATA UNA STRINGA
CONTA QUANTI CAR. NON-ASCII

```
func contaNonASCII (s string) int {
    count:=0
    for _, c := range s {
        if c > 127 {
            count++

        }

    }
    return count

}
```

```
func    farfallino(s  string) string {
    var  t  string

    for  _, c := range  s   {
        if  c=='a' || c=='e' ||
            c=='i' || c=='o' ||
            c=='u'  {
                t +=string(c)+"f"+string(c)
        } else {
                t+= string(c)
        }
    }
    return t
}
```

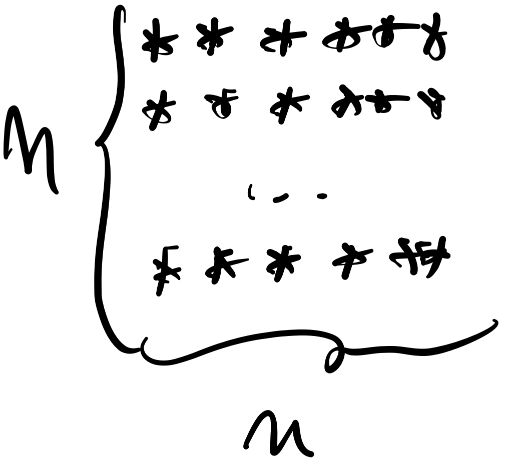
URL



```
func getHost(url string) string {
    bb := strings.Index(url, "//")
   b := strings.Index(url[bb+2:], "/")
                                  +bb+2
    return url[bb+2 : b]
}
```

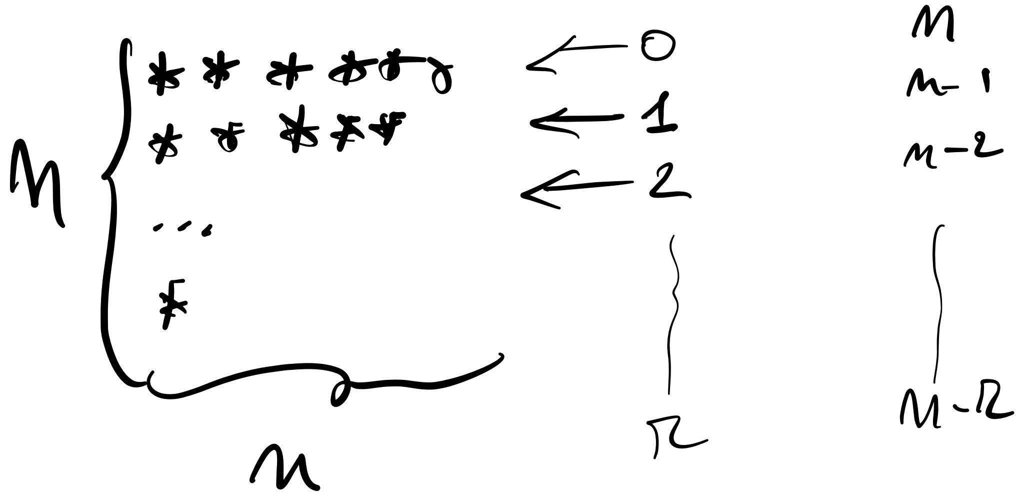
PATTERN

- Scrivere un programma che dato n stampa



```
var n int
fmt.Scan(&n)
for r := 0; r<n; r++ {
    for c := 0; c<n; c++ {
        fmt.Print("*")

    }
    fmt.Println()
}
```

- Scrivere un programma che dato n stampa



```
var n int
fmt.Scan(&n)
for r:=0; r<n; r++ {
        for c:=0 ; c<n-r ; c++{
                fmt.Print("*")
        }
    fmt.Println()
}
```