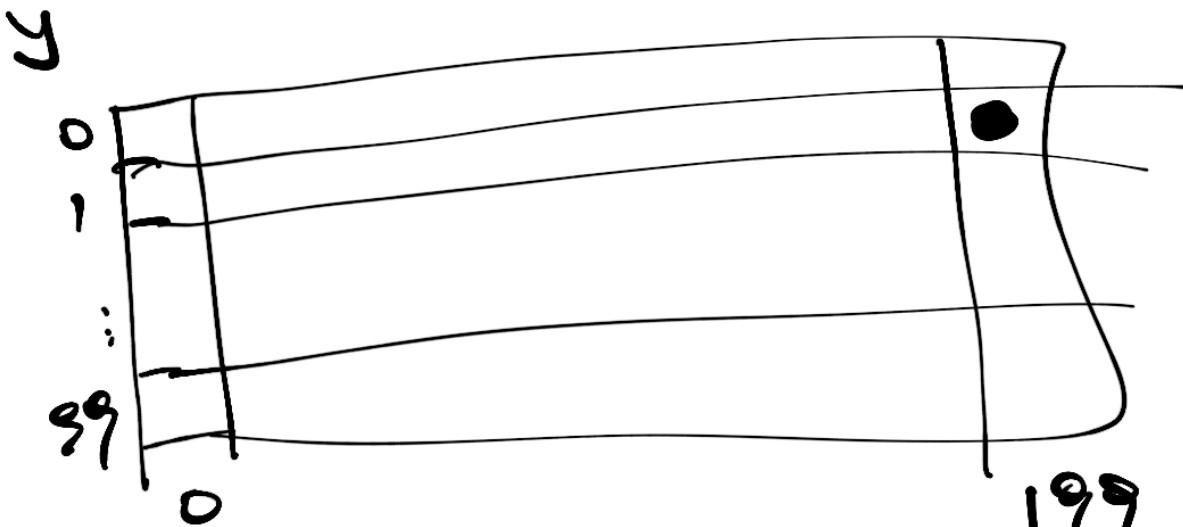


ARRAY / SLICE MULTIDIMENS.

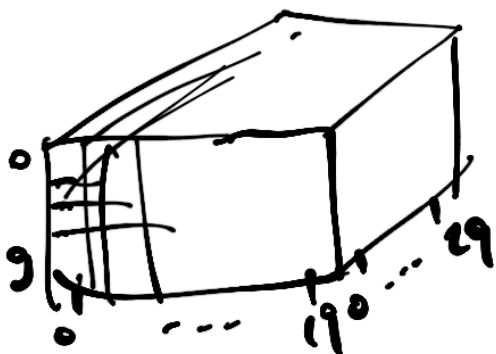
var x [5][7] int
y [100] [200] string

y[i][j]



z [10] [20] [30] float

z[i][j][k]



var x [][]string

x = append(x, []string {"a", "b", "c"})

x = append(x, []string {"un", "wer",
"mer", "gio", ..., "don"})

x[i][j]



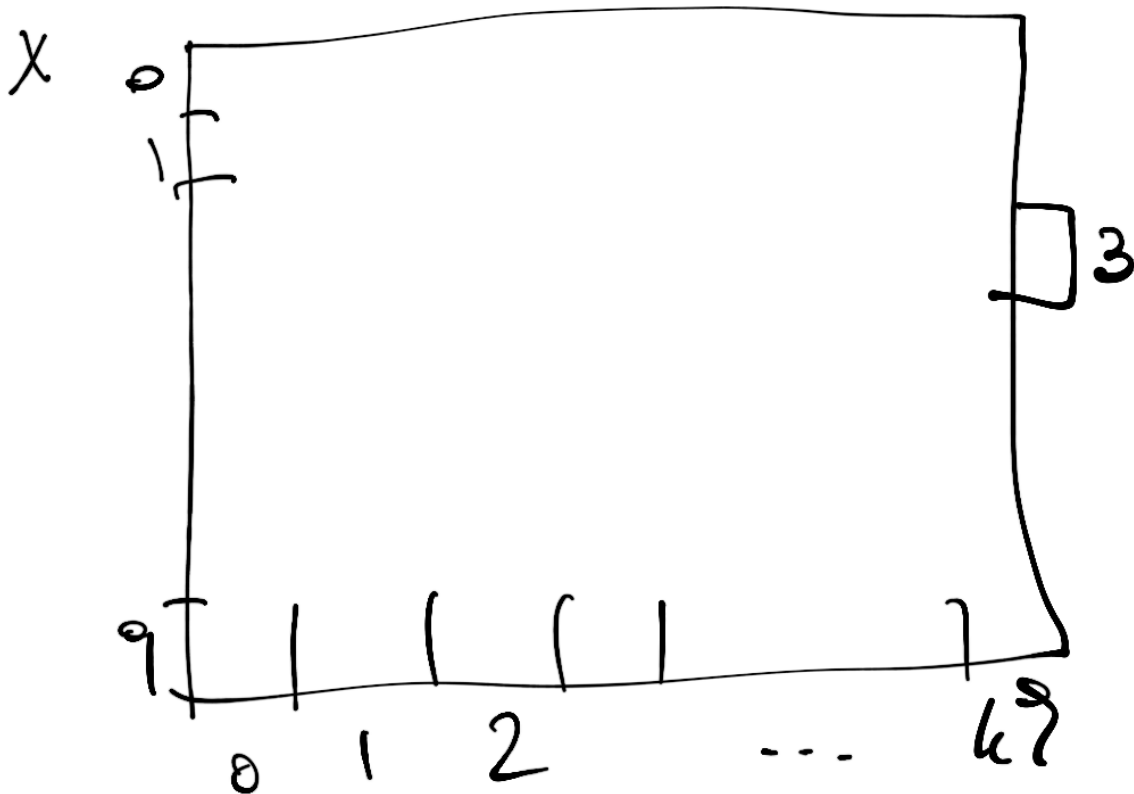
var x [][] string

for i := 0; i < 10; i++ {

 x = append(x, make([]string, 5))

}

 x[3] = append(x[3], "--")



var a [10][50] string

RICORSIONE

- 1) MEMORIA STATICA
- 2) STACK
- 3) HEAP

① VARIABILI GLOBALI

③ GLI OGGETTI ALLOCATI
CON new/make

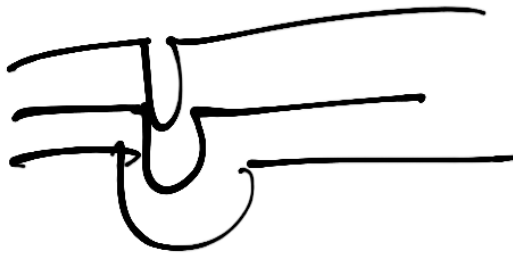
② VIVONO LE VARIABILI
LOCALI

STACK

LIFO

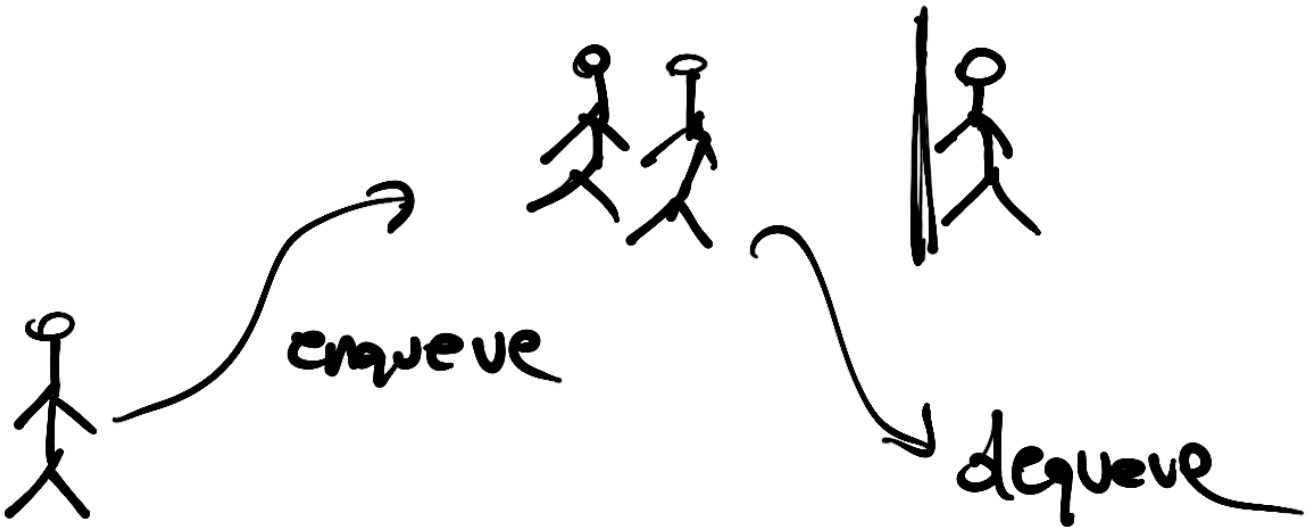
push

pop



QUEUE

FIFO



func f (a, b int) (c int) {

var x, y int
var z *int

z = new (int)

var s []string

s = make ([]string, 5)

}



RECORD DI
ATTIVAZIONE

(di una funzione)

V. LOCALI

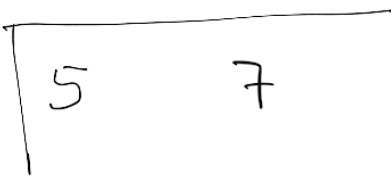
VALORI RESTITUITI

PUNTO DI RITORNO

```
1 func main() {  
2   var x, y, r is int  
3   fmt.Scan(&x)  
4   fmt.Scan(&y)  
5   r = f(x, y)  
6   fmt.Printf("%d\n", r)  
7 }
```

```
8 func f(a, b int)(c int) {  
9   var x, y int  
10  x = sqr(a)  
11  y = sqr(b)  
12  c = x + y + 1  
13  return  
14 }
```

```
15 func sqr(x int)(z int) {  
16  z = x * x  
17  return  
18 }
```



RICORSIONE

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (n-1) \cdot n$$

$$\underline{n!} = \begin{cases} 1 & \text{se } n=0 \\ n \cdot \underline{(n-1)!} & \text{altrimenti} \end{cases}$$

$$\begin{aligned} 5! &= 5 \cdot 4! = 5 \cdot 4 \cdot 3! = 5 \cdot 4 \cdot 3 \cdot 2! = \\ &= 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 \cdot 0! = \\ &= 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 \cdot 1 \end{aligned}$$

itopinonavevanonipoti
omordotuanuoraoarounauto dromo

1) le parole con meno
di due lettere sono
palindrome

2) una parola con più
di una lettera è
palindroma se e solo
se è della forma

a | W | a

e W è palindroma

$$n! = 1 \cdot 2 \cdot 3 \dots \cdot n$$

$$n! = \begin{cases} 1 & \text{se } n=0 \\ n \cdot (n-1)! & \text{altrimenti} \end{cases}$$

```

func fatt (n int) (ris int) {
  ris=1
  for i:=1; i<=n; i++ {
    ris*=i
  }
  return
}

```

```

func fatt (n int) int {
  if n==0 {
    return 1
  }
  else {
    return n * fatt(n-1)
  }
}

```

-SCRIVETE UNA FUNZIONE
isPalindroma che dica
SE UNA STRINGA ASCII
È PALINDROMA

USANDO LA DEF.

RICORSIVA

func isPalindroma (s string) bool {
 if len(s) <= 1 {
 return true

}
 if s[0] != s[len(s)-1] {
 return false

}
 return isPalindroma (s[1:len(s)-1])

}

var s string

t := []rune(s)

```

func isPalindrom (s string) bool {
  t := []rune(s)
  if len(t) <= 1 {
    return true
  }
  if t[0] != t[len(t)-1] {
    return false
  }
  return isPalindrom(string(
    t[1: len(t)-1]))
}

```

}

isPalindrom("a")

NUMERI DI FIBONACCI

1 2 3 5

F_0	F_1	F_2	F_3	F_4	F_5
1	1	2	3	5	8
F_5	F_6	F_7	...		

$$\lim_{n \rightarrow \infty} \frac{F_{n+1}}{F_n} = \rho$$

$$\rho = \lim_{n \rightarrow \infty} \frac{F_{n+1}}{F_n} = \lim_{n \rightarrow \infty} \frac{F_n + F_{n-1}}{F_n} =$$

$$= 1 + \lim_{n \rightarrow \infty} \frac{F_{n-1}}{F_n} = 1 + \frac{1}{\rho}$$

$$\rho = 1 + \frac{1}{\rho}$$

$$\rho^2 = \rho + 1$$

$$\rho^2 - \rho - 1 = 0$$

$$\rho_{1,2} = \frac{1 \pm \sqrt{1 + 4}}{2} = \frac{1 \pm \sqrt{5}}{2}$$