var p *int

var x int

p = &x

*p = 7 } fmt.Printlln(*p)

GESTIONE DELLA MEMORIA

1) MEMORIA STATICA
2) STACK
3) HEAP

---

package main

var x int

func f() {

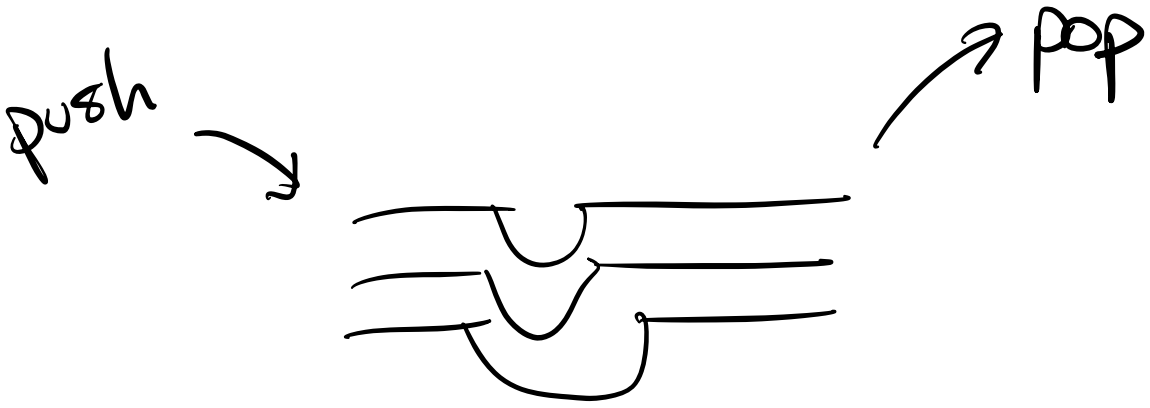
}

HEAP

- new
- make

STACK

LIFO



STACK DI ESECUZIONE

RECORD DI ATTIVAZIONE

PUNTO DI RIENTRO
VAR. LOCALI
VALORI RESTITUITI

```
1  func main() {
2      var x, y, ris int
3      fmt.Scan(&x)
4      fmt.Scan(&y)
5      ris = f(x, y)
6      fmt.Printf("%d\n", ris)
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
13     return
14 }
15 func sqr(x int) (a int){
16     a = x * x
17     return
18 }
```

RICORSIONE

$n!$

$$n! \triangleq 1 \cdot 2 \cdot 3 \cdot \ldots \cdot n$$

CASO BASE

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

$$3! = 3 \cdot 2! =$$
$$= 3 \cdot 2 \cdot 1! =$$
$$= 3 \cdot 2 \cdot 1 \cdot 0! =$$
$$= 3 \cdot 2 \cdot 1 \cdot 1$$

$$n! \triangleq \begin{cases} 1 & \text{se } n=0 \\ 2n + (n+1)! & \text{altrimenti} \end{cases}$$

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

casacca

ANAGRAMMI

cane

cane
caen
neac
neca
⋮

anagrams("cane") =

"c" + X $\quad \forall x \in$ anagrams("ane")

"a" + X $\quad \forall x \in$ anagrams("cne")

"n" $\quad$ ...

"e"