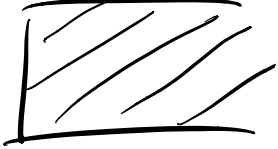


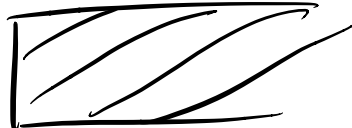
0-ario

for {
[
}]



1-ario

for {cond} {
[
}]



for delta := $b + b - 4 * a * c$; delta > 0; {

...

}

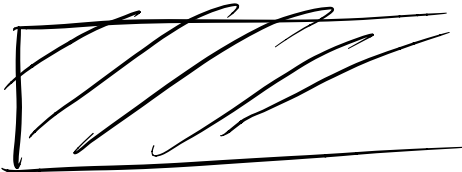
3-ario

A

inizializzazione
di variabile

for B {

condizione



C

aggiornamento

}



for {A}; {B}; {C} {



}

```

p := 1
for p <= n {
    fut. Println(p)
    p *= 2
}

```



```

for p := 1; p <= n; p *= 2 {
    fut. Println(p)
}

```

```

for i {
    fut. Println("ciao")
}

```

$n=5$

```

var i int
for i = 0; i < n; i++ {
    fut.Println("#")
}
  
```

↑
IL CICLO PARADIGMATICO

(B)

```

for i = 1; i <= n; i++ {
    fut.Println("#")
}
  
```

(C)

```

for i = n; i > 0; i-- {
    fut.Println("#")
}
  
```

(D)

```

for i = n-1; i >= 0; i-- {
    fut.Println("#")
}
  
```

- Chiede per prima cosa quanti bambini ci sono
- chiede di inserire le loro altezze
- Stampa l'altezza media

Stabilire se un numero è primo

```

var n, i int
fun Scan (&n)
  for i = 2; i < n; i++ {
    if n % i == 0 {

```

break

```

    }
  }
  if i < n {
    fun Println("Non è primo")
  } else {

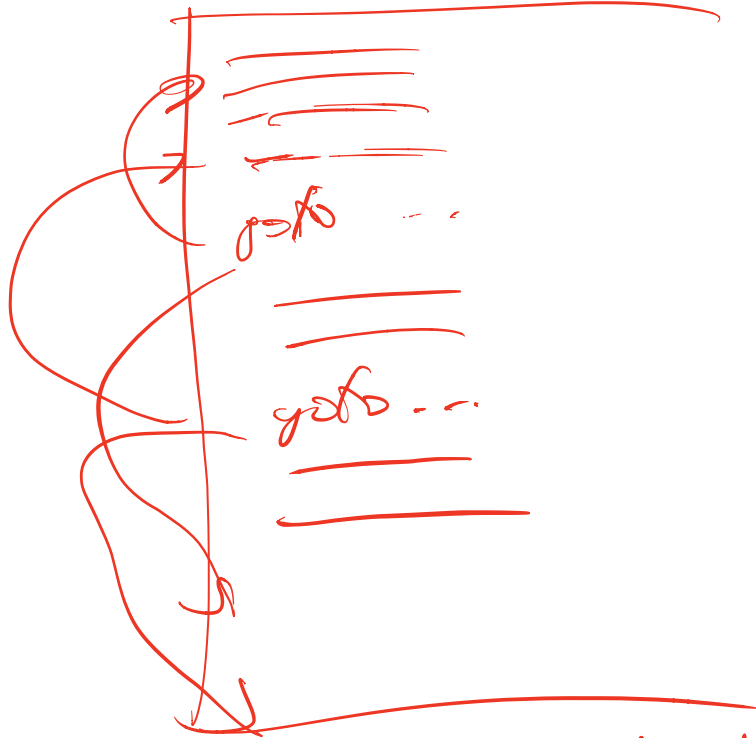
```

se sono uscito prematuramente

fat. Print ("E prho")

}

go to



"spaghetti programming"

Dijkstra

« On goto considered
harmful »

Knuth

« Structured programming
with goto's »

«The Art of Computer Programming»

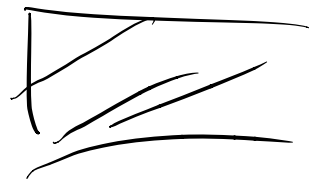
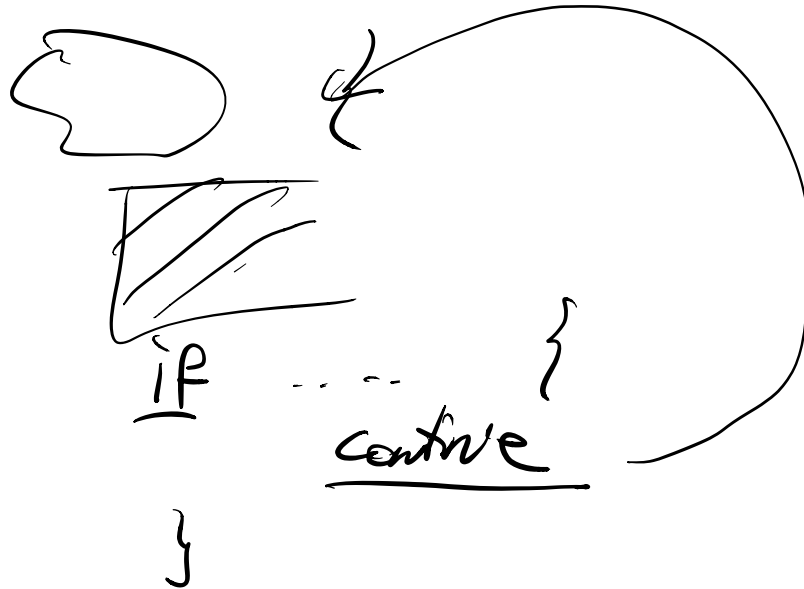
TACP

break / continue

interrompe

forza la prossima
esecuzione del
ciclo

for



}