```
var x, y, m int
fmt.Scan(&x, &y)
if x < y {
    m = x
} else {
    m = y
}
for x%m != 0 || y%m != 0 {
    m--
}
fmt.Println(m)
```

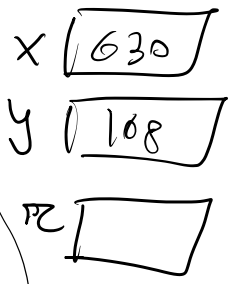
# ALGORITMO DI EUCLIDE PER IL MCD

1 — Dati due numeri interi positivi $x$ e $y$

2 — $r \leftarrow x \,\%\, y$

3 — Se $r \neq 0$
$x \leftarrow y$
$y \leftarrow r$
go to 2

4 — Il MCD è $y$




630
108

Knuth

```
var x, y, r int
fmt.Scan(&x, &y)
r = x % y
for r != 0 {
    x, y = y, r
    r = x % y
}
fmt.Println(y)
```

time ./mcol
7
17

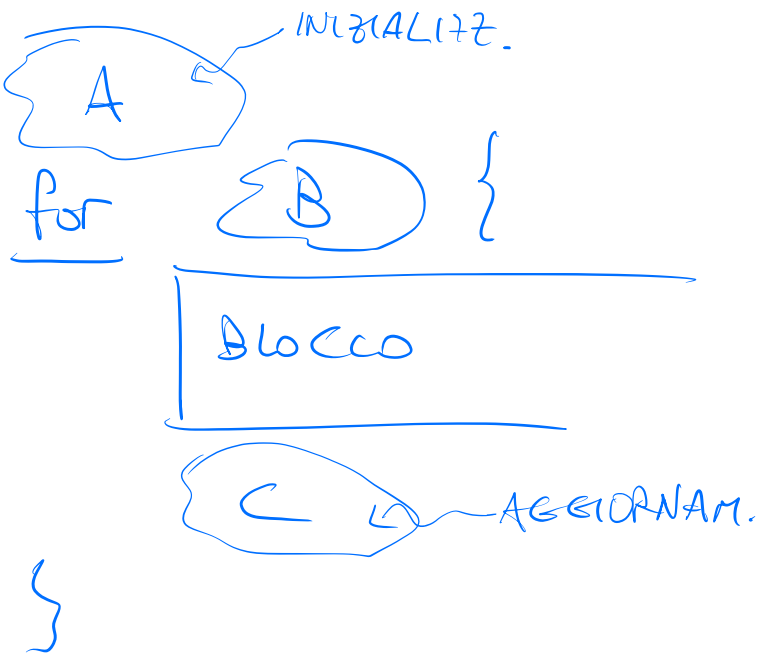
FOR 0-ARIO

CICLO INFINITO

```
for {
    BLOCO
}
```

FOR 3-ARIO

```
for (A ; B ; C) {
    BLOCO
}
```




```
var  x, y, r   int
fmt.Scan(&x, &y)
r = x % y
for  r != 0 {
     x, y = y, r
     r = x % y
}
fmt.Println(y)
```

```
for r = x % y; r != 0; r = x % y {
    x, y = y, r
}
```

```
S := 0
for i := 1; i <= n; i++ {
    S += i
}
```

---

① `for i := 0; i < 10; i++ { fmt.Println("*") }` — CICLO PARADIGMÁTICO

② `for i := 1; i <= 10; i++ { fmt.Println("*") }`

③ `for i := 10; i > 0; i-- { fmt.Println("*") }`

Stampa dei primi n quadrati
$(1^2, 2^2, 3^2, \ldots)$

```
var n int
fmt.Scan(&n)
for i := 1; i <= n; i++ {
    fmt.Println(i*i)
}
```

var x, y, z int

fmt.Scan(&x, &y, &z)

if x < y < z {  → x < y && y < z

} else if x < z < y {

} else if

---

if x <= y {

if y < z

else if z <= y

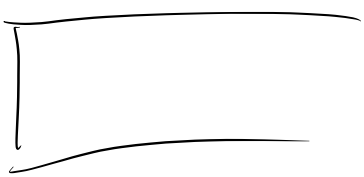
}

var n int

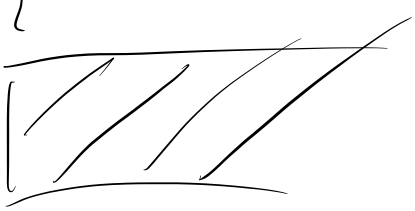
fmt.Scan(&n)

if n>0 && n<10 {

} else if n<100 {



} else {



}

n%10 (n/10)%10 (n/100)%10

if (n%10)%2==1 && ((n/10)%10)%2==1

&& ((n/100)%10)%2==1

# ESEMPI DI LETTURE ITERATE DI INPUT

④ NUMERO DI INPUT NOTO

③ NUMERO DI INPUT INCOGNITO

④

```
var n,alt,s int
fmt.Scan(&n)
for i:=0; i<n; i++{
    fmt.Scan(&alt)
    s+=alt
}
fmt.Println(float64(s)/float64(n))
```

(B)

```
var n, alt, s int
fmt.Scan(&alt)

for alt != 0 {
    n++
    s+=alt
    fmt.Scan(&alt)
}
fmt.Println(float64(s)/float64(n))
```

## break

- Istruzione che forza l'uscita da un ciclo

```
var n, alt, s int
for {
    fmt.Scan(&alt)
    if alt == 0 {
        break
    }
    n++
    s += alt
}
fmt.Println(float64(s)/float64(n))
```

## continue

- Forza la prossima esecuzione del ciclo

```
var n, alt, s int
for {
    fmt.Scan(&alt)
    if alt == 0 {
        break
    }
    if alt < 0 {
        continue
    }
    n++
    s += alt
}
fmt.Println(float64(s)/float64(n))
```

- Stabilire se un numero è primo

```
var found bool
var x int
fmt.Scan(&x)
for d := 2; d < x; d++ {        // !found
    if x % d == 0 {
        found = true
        break
    }
}
if found {
    fmt.Println("Non è primo")
} else {
    fmt.Println("È primo")
}
```

$\forall d = 2, \dots, x-1$