

```
var x, d int
func Scan (&x)
```

```
for d=2; d<x; d++ {
    if x%d == 0 {
```

```
        break
```

```
    }
```

```
    if d < x {
        fmt.Println ("Composite")
```

```
    } else {
        fmt.Println ("Primo")
```

```
    }
```

⊛

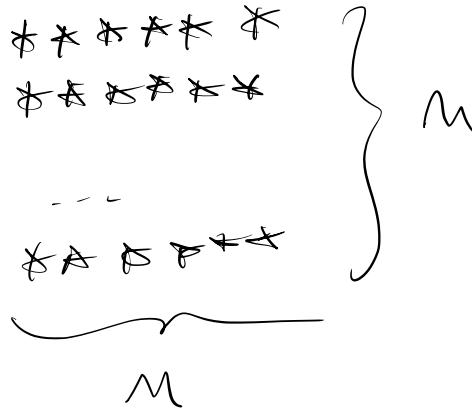
```
var x, d int  
for {  
    fun. Scan (&x)  
    if x == 0 {  
        break  
    }  
}
```

```
for d=2; d<x; d++ {  
    if x/d == 0 {  
        break  
    }  
    }  
if d < x {  
    }  
else {  
    }  
}
```

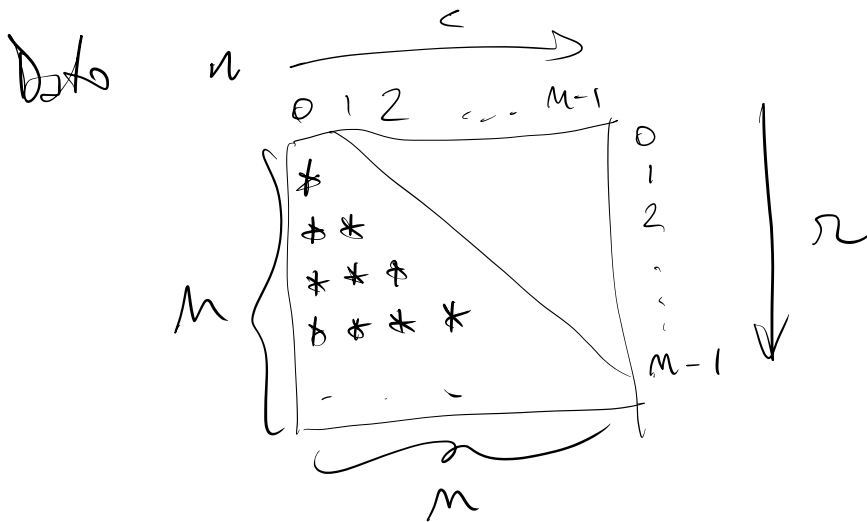
```
}
```

Esercizio

Dato n stampa



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



```

var m int
fun Scan (&m)
for r:=0; r<m; r++ {
  for c:=0; c<r+1; c++ {
    fun.Print("*")
  }
  fun.Println()
}

```

```

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

```

package main

import "fmt"

func isPrime (x int) bool {

PARAMETRI
FORMAZI

TIPO
RISPOSTA

INTERAZIONE

var d int

for d = 2; d < x; d++ {

if x % d == 0 {
break

}

if d < x {
return false

} else {
return true

}

return d == x

func

main () {
var n int

fmt.Scan(&n)

if isPrime(n) {
fmt.Println("È primo")

} else {
fmt.Println("È composto")

CHIAMATA
DELLA
FUNZIONE

PARAMETRI
ATTUALI
(ARGOMENTI)

ARE KONTAK

```
func isPrime(x int) bool {  
  var d int  
  for d=2; d<x; d++ {  
    if x/d == 0 {  
      return false  
    }  
  }  
  return true  
}
```

PARAMETRI ATTUALI vs. FORMALI

PAR. ATTUALI = ESPRESSIONE

```
if isPrime(x*x + y) {  
  ...  
}
```

1E7

~~1E7~~

```
func   sqr(x int) int {  
    return x*x  
}
```

...

$$\text{delta} = \text{sqr}(b) - 4*a*c$$

if isPrime(sqr(x)+y) ...

Scrivere una funzione pow
che dati du e inter; a e B
restituisce a^B .

pow(x,y int)


```

func pow (x int, y int) int {
    ris := 1
    for i := 0; i < y; i++ {
        ris *= x
    }
    return ris
}

```

}

```

func main () {
    var a, b int
    fmt.Scan (&a, &b)
    fmt.Println (pow (a, b))
}

```

}

if

```

isPrime (pow (x,  $x^{(y+1)^2}$ ) &&
         pow (x, 3) = 7 ...
          $x^3$ )

```

```
2.go
func main() {
    --
}
```

```
b.go
func pow {
    --
}
```

go build 2.go b.go -o p110.exe