

```
func isPrime(x int) bool {  
    if x <= 1 return false;  
    for i := 2; i < x; i++ {  
        if x % i == 0 {  
            return false  
        }  
    }  
}
```

```
    }  
    return true
```

}

func main() {  
    var n, x, c, y int

    fmt.Scan(&n)

for x, c = 2, 0; c < n; x \*= 2 {  
        → if isPrime(x-1) {

            y = x - 1  
            c++

        }

    }

    fmt.Println(y)

}

- Data una stringa che rappresenta un numero in base 2, calcolare il numero.

COMMA-OR IDIOM  
COMMA-ERROR IDIOM

```
func binaryToInt(s string) (int, bool) {  
    if len(s) == 0 {  
        return 0, false  
    }  
    v := 0  
    for _, r := range s {  
        switch r {  
            case '0': v *= 2
```

var v  
case '1' :  $v = 2 * v + 1$   
default :  
return 0, false

}  
}  
return v, true

}

x, \_ := binaryToInt("10110")

var s string  
fmt.Scan(&s)  
x, ok := binaryToInt(s)  
if ok {

```
-  
    fut.Println("I love  
        e", x)  
    } else {  
        fut.Println("Error  
            nell'input")  
    }  
}
```

## strconv

• `Atoi (s string) (int, error)`  
    ↓  
    `== nil`    OK!  
    `!= nil`    ERR!

`x, err := Atoi ("537")`

`x, err := Atoi ("6e40")`

if `err != nil` ...

• `Itoa (x int) string`

• `ParseInt (s string, base int, bitSize int) (int, error)`

$x, ok := \text{binaryToString}(s)$



$x, err := \text{strconv.ParseInt}(s, 2, 64)$

$xInt := \text{int}(x)$

if  $err \neq nil$  {

$\text{fmt.Println}("ERROR")$

{ else }

$\text{fmt.Println}(xInt)$

}

NUMERO SEGRETO

##3##4##1###5

RISULTATO:  $\frac{S}{5}$  il doppio del  
numero segreto

```
func secret(s string) (int, bool) {  
    var t string  
    for _, r := range s {  
        if unicode.IsDigit(r) {  
            t += string(r)  
        } else if r != '#' {  
            return 0, false  
        }  
    }  
    x, err := strconv.Atoi(t)
```



if err != nil ?  
return 0, false

}  
return 2\*x, true

}

PER CASA

\* \* \* \* \*

\* \* \* \* \*

\* - , err := strconv.Atoi(string(r))  
if err == nil {  
 t += string(r)

{ else

|           |       |     |       |
|-----------|-------|-----|-------|
| CONTAU    | VOCAU | E   | CONS. |
| MINUSCULE | IN    | UAA |       |
| STRINGA   | ASCII |     |       |

func conta (s string) (cv int, cc int) {

cv = 0

cc = 0

for i := 0; i < len(s); i++ {

switch s[i] :

case 'a', 'e', 'i', 'o',  
'u' :

cv++

case 'b', 'c', ... :

cc++

}  
return

}

- DATA UNA STRINGA E  
UN INTERO, FARE UNO  
SHIFT

carssd 3

csss car  
programmazione 5

lunazione progr