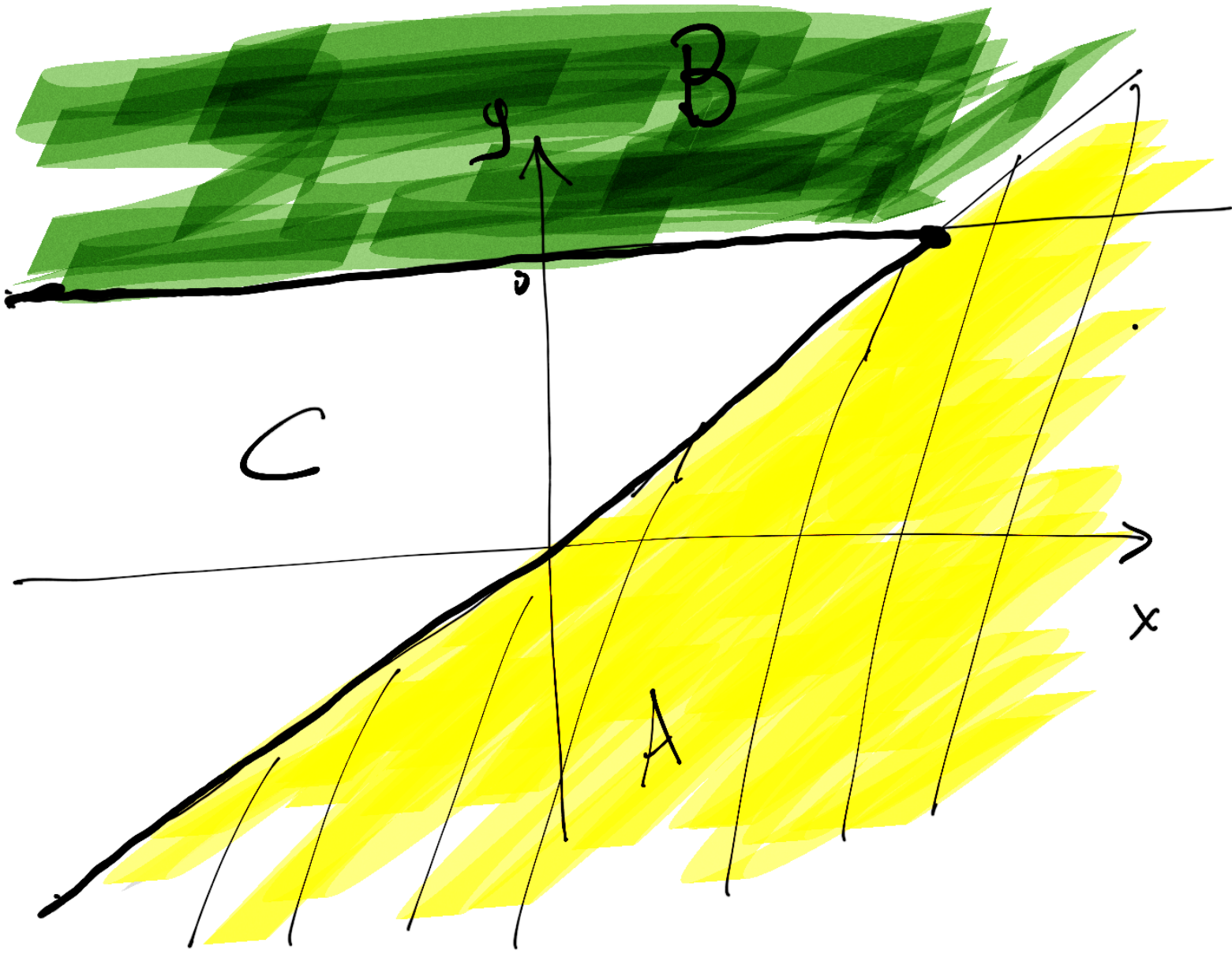
switch senza selettore

```
switch {
    ---
    ---
    ---
}
```

≡

```
switch true {
    ---
    ---
    ---
}
```

```
var x,y float64
...
switch {
        case x>y :
                fmt.Println("A")
        case y>5 :
                fmt.Println("B")
        default:
                fmt.Println("C")
}
```


B
y
C
x
A

FUNZIONI

SEGNATURA

IDENTIFICATORE

func nomeFunzione (parametriFormali) tipiRest {

}

---

parametriFormali

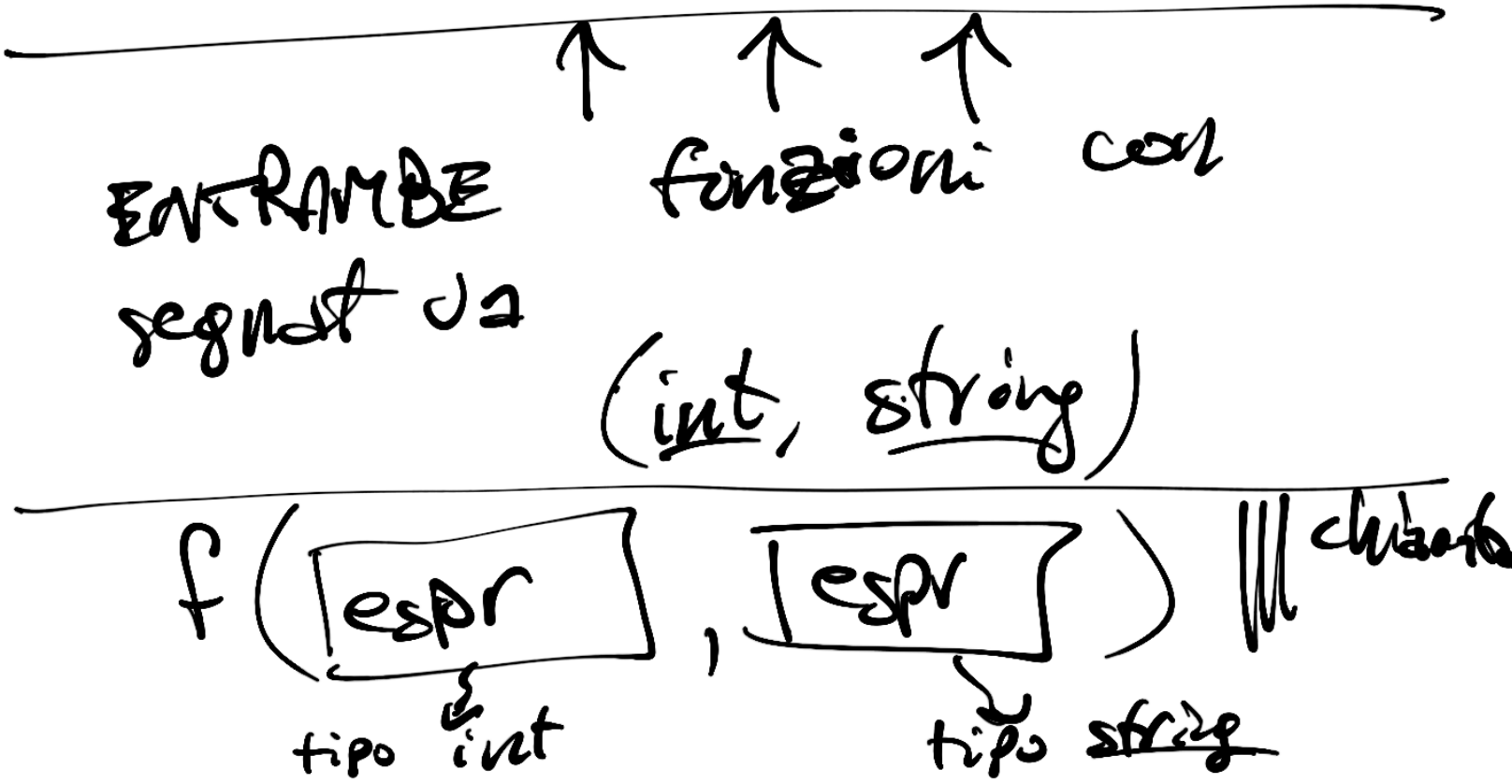
elenco (separato da ,) di "dichiarazioni di variabili"

tipiRest

elenco di tipi di valori che vengono restituiti

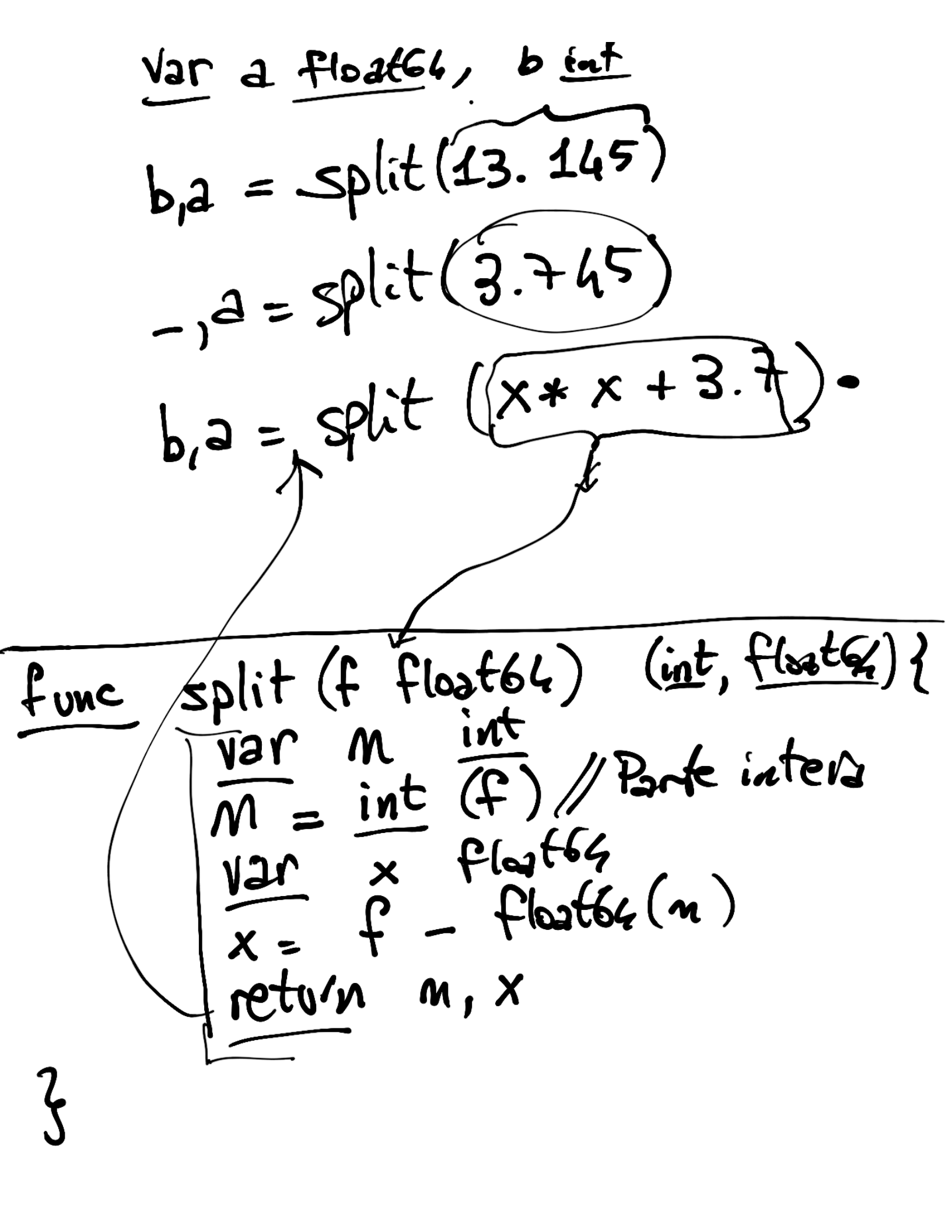
func f (x int , s string)

func g (y int , pippo string)



```
func nextCollatz(x int) int {
    if x%2 == 0 {
        return x/2
    } else {
        return 3*x+1
    }
}

func main() {
    var n int
    fmt.Scan(&n)
    for n != 1 {
        fmt.Println(n)
        n = nextCollatz(n)
    }
    fmt.Println(1)
}
```


var a float64, b int
b,a = split(13.145)
_,a = split(3.745)
b,a = split(x*x+3.7)
func split(f float64) (int, float64) {
var n int
n = int(f) // Parte intera
var x float64
x = f - float64(n)
return n, x
}

```
// This function determines if
// x is a power of 2

func isPowerOfTwo(x int) bool {
    for x > 1 {
        if x%2 != 0 {
            return false
        }
        x /= 2
    }
    return true
}
```

pot.go

```
func main() {
    var n int
    fmt.Scan(&n)
    fmt.Println(isPowerOfTwo(n))
    fmt.Println(n)
}
```

main.go

main.go

```
package main
import "fmt"
func main() {
    ...
}
```

pot.go

```
package main
func powerOfTwo(...
    ...
}
```

```
$> go build pot.go main.go
```

UN PRIMO DI MERSENNE
È UN NUMERO CHE

1) è primo
2) è della forma

$2^p - 1$

---

3 7 31 ...

SCRIVETE UNA FUNZIONE
CHE STABILISCE SE UN
NUMERO È UN PRIMO
DI Mersenne

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

```
func isMersennePrime(x int) bool {
    return isPrime(x)&&
           isPowerOfTwo(x+1)
}
```

```
func main () {
    var n int
    fmt.Scan (&n)
    for x:=2; x<n; x++ {
        if isMersenne Prime (x){
            fmt.Println (x)
        }
    }
}
```

```
time ./main
1000 000 000
```

SCRIVETE UNA FUNZIONE CHE DATO UN INTERO x RESTITUISCA IL PIÙ PICCOLO $y > x$ CHE SIA UN PRIMO DI MERSENNE

```
func contaVC(s string) (int, int) {
    var voc, cons int
    for _, c := range s {
        switch {
            case c=='a', c=='e', ..., c=='u':
                voc++;
            case unicode.IsLetter(c):
                cons++;
        }
    }
    return voc, cons
}
```

```
var a, b int
a, b = contaVC("Ciao mondo")
```

```
func contaVC(s string) (voc int, cons int){

    for _, c := range s {
        switch {
            case c=='a', c=='e', ..., c=='u':
                voc++;
            case unicode.IsLetter(c):
                cons++;
        }
    }
    return
}
```