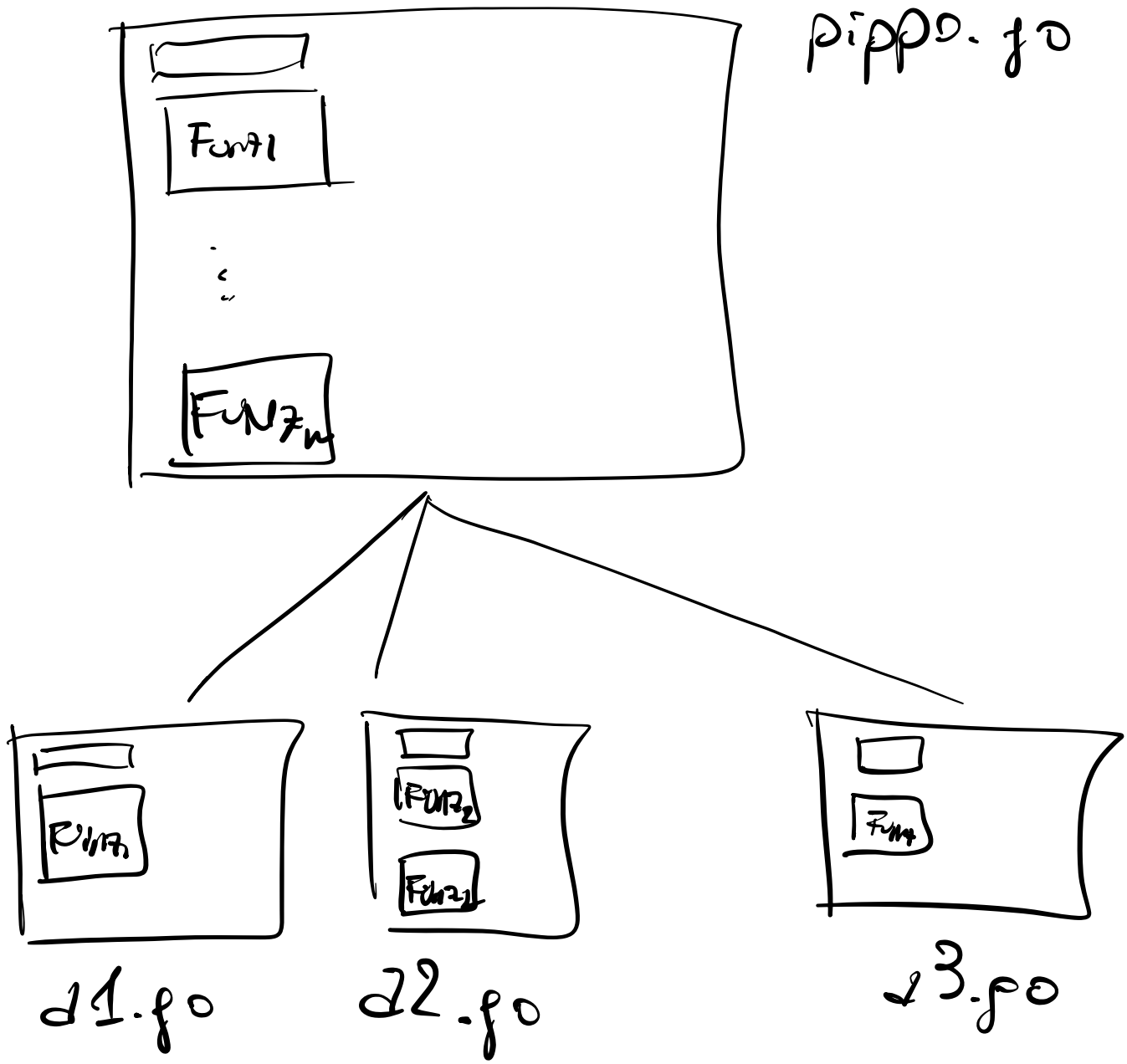
FUNZIONI

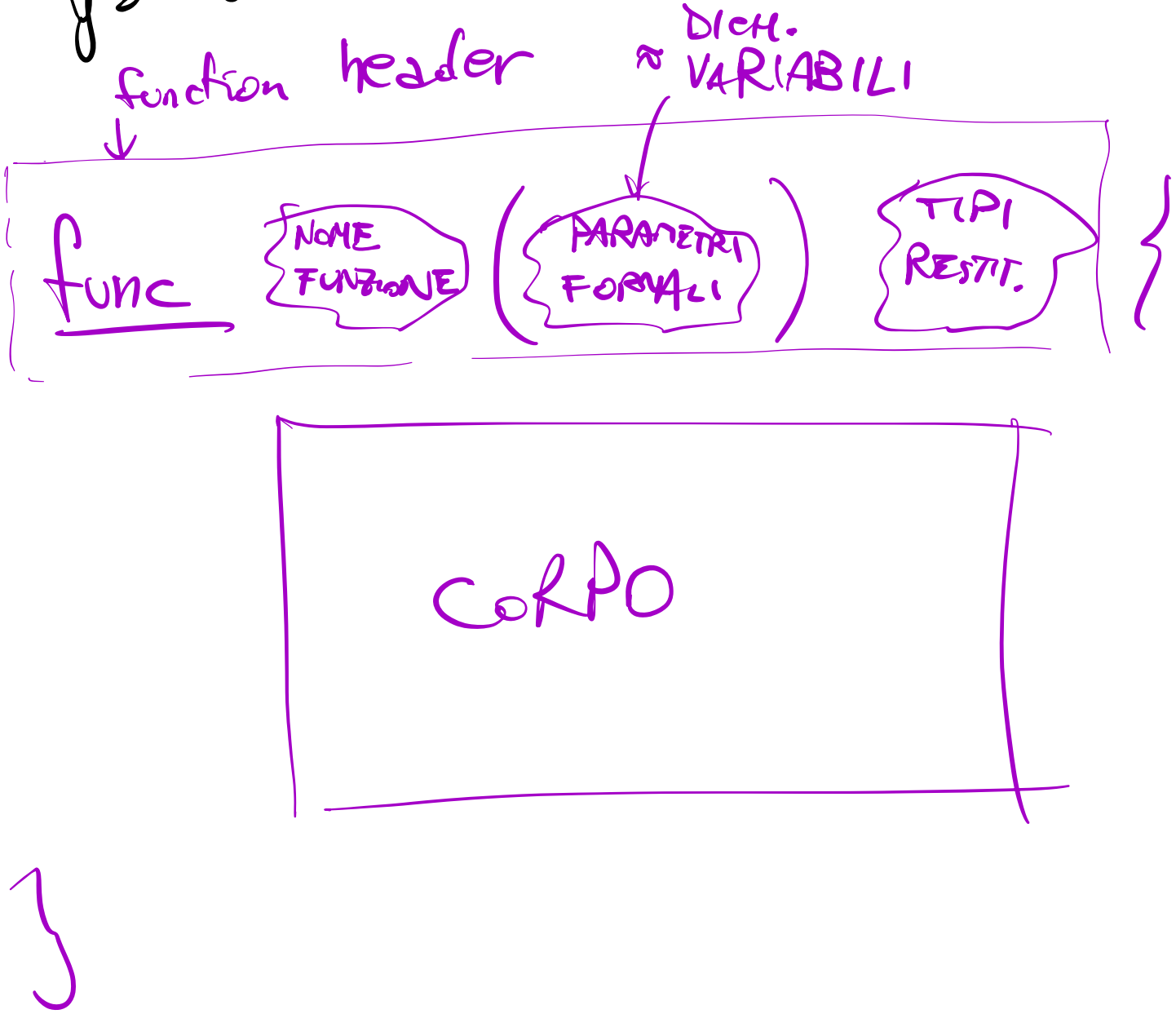
SORGENTE GO ≡ COLLEZIONI DI FUNZIONI + ...



go build f1.go f2.go f3.go

-o risultato

go build -o risultato *.go



}

```
func pippo (a int) ...
func pluto (a int, b string) ...
func minnie (a, b string, c int) ...
```

```
func gino(...) {
func anna(...) int {
func beppe(...) (int, string) {
func paolo(...) (int, int, float64) {
```

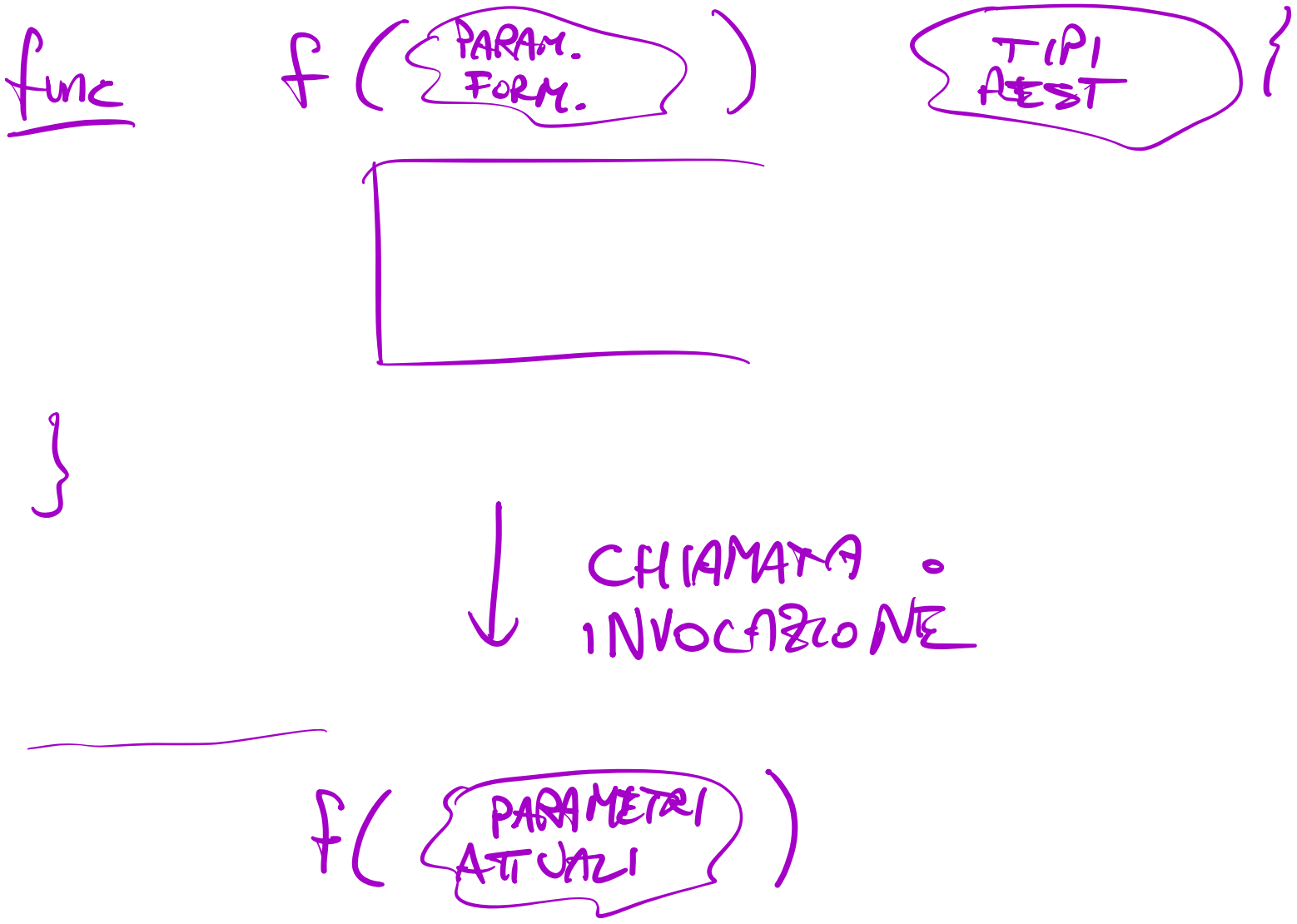
return espr1, ..., esprn

```
func anna(...) (int, string) {
    if (_____) {
        return 0, "ciao"
    }
    ===
    return 3, "pippo"
}
```

NOMI AI VALORI RESTITUITI

```
func separa(x float64) (int, float64) {
    pi := int(x)
    pf := x - float64(pi)
    return pi, pf
}

func separa(x float64) (pi int, pf float64) {
    pi = int(x)
    pf = x - float64(pi)
    return
}
```


func
f (PARAM. FORM.)
TIPI REST
{
}
CHIAMATA o INVOCAZIONE
f(PARAMETRI ATTUALI)

func pippo (a int) ...

func pluto (a int, b string) ...

func minnie (a, b string, c int) ...


```
pippo (3)
pippo ( x + 3 )
pluto ( int(x) + y, "ciao" + z )
pippo (a)
pippo (x)
```

ESERCIZI

1) Un primo di Mersenne è un primo della forma

$$2^p - 1 \quad (\text{con } p \text{ primo})$$

Scrivete due funzioni
- Una stabilisce se un numero è un primo di Mersenne
- Una restituisce il k-esimo primo di Mersenne (dato k)

2) Scrivete una funzione che data una stringa restituisca il numero di vocali e il numero di consonanti

3) Data una stringa così

ci sono ** coccodrilli
ed * orangotango,
** piccoli serpenti ...

restituisce

ci sono 2 coccodrilli
ed 1 orangotango,
2 piccoli serpenti ...

ASSEGNI

EUR 35. 15

TRENTACINQUE, 15

20 → 99

23 ventitre

77 settanta sette

82 ottanta due

21 ventuno

28 vent otto

```
// x compreso tra 1 e 9
func unita (x int) string {
    switch (x) {
        case 1:  return "uno"
        case 2:  return "due"
        :
        case 9:  return "nove"
        default: return "CANTHAPPEN"
    }
}
// x compreso tra 2 e 9
// estrae la decina (e.g. "vent")
// e la vocale finale (e.g. 'i')
func decina (x int) (s string, c rune) {
    switch (x) {
        case 2:  s = "vent"
                 c = 'i'
        case 3:  s = "trent"
                 c = 'a'
        ...
```

```
        case 9:  s = "nove"
                 c = 'a'

    }
    return

}

//x compreso fra 0 e 99
func fra0e99(x int) string {
    switch (x) {
        case 0 : return "zero"
        case 10 : return "dieci"
                :
        case 19 : return "diciannove"
    }
    if x < 10 {
        return unita(x)
    } else {
        d := x/10
        u := x%10
        dec, voc := decine(d)
```

```
uni:= unita(u)
if u==1 || u==8 {
        return dec+uni
} else {
        return dect+string(voc)+
               uni
}
}
}
// x compreso fra 0 e 999
func fra0e999(x int) string {
    var cent string
    if x>=100 && x<=199 {
        cent= "cento"
    }else if x>=200 {
        cent= unita(x/100) + "cento"
    }
    if x%100 == 0 && x!=0 {
        return cent
```

```
    }
    return cent + fra0e99(x / 100)
}
```

ESERCIZIO

Data una stringa convertire i numeri in lettere:

ci sono 12 coccodrilli
e 372 orangotango

↓

ci sono dodici coccodrilli
e trecentosettantadue
orangotango