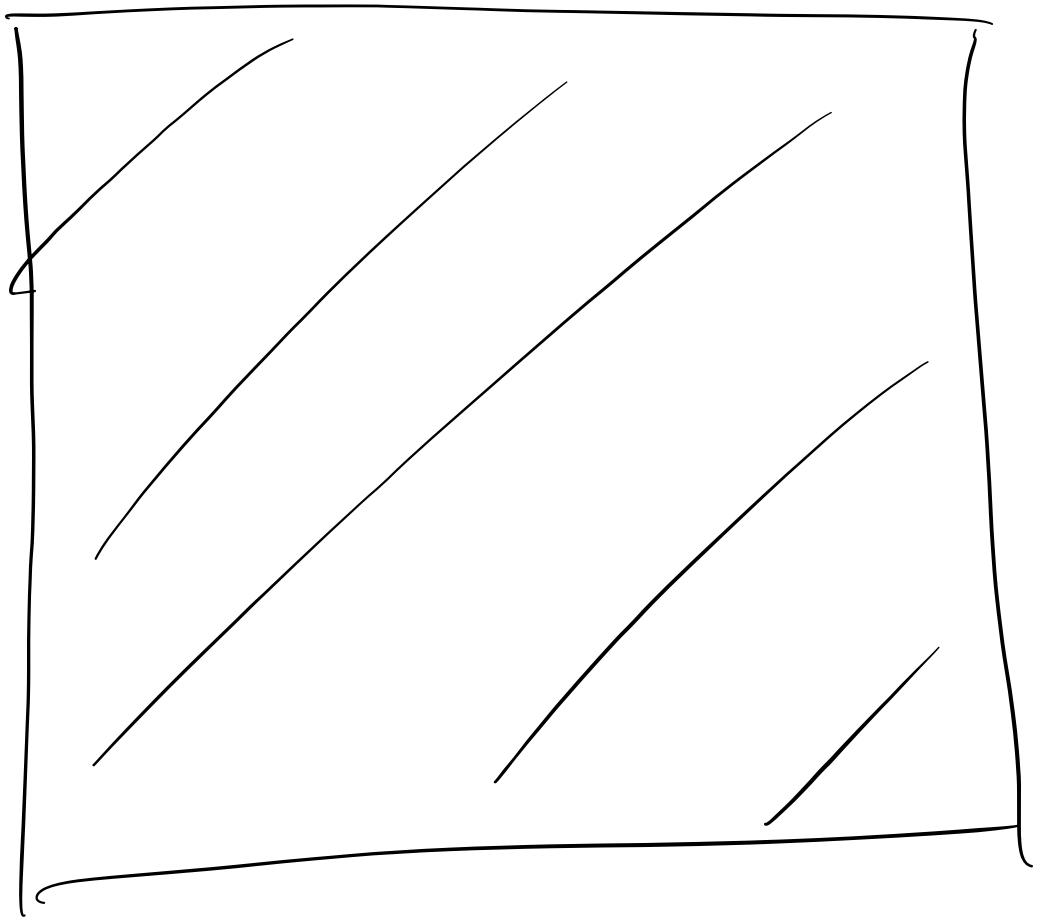
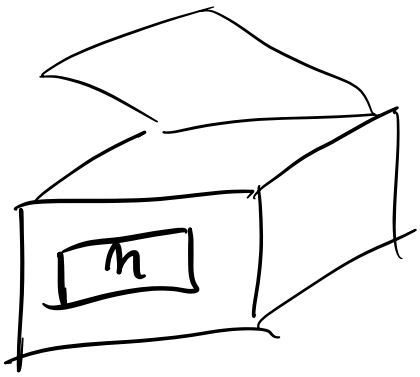
```
package main
import (
    "fmt"
    "   "
    ...
)
func main() {
```



```
}
```

VARIABILI



identificatore

VARIABILE
- NOME
- TIPO
- SCOPE

} statiche

- VALORE

} dinamica

DICHIARAZIONE DI VARIABILE

var {id, id, id} {tipo}

(A)

```
var x, y int
var f float64
```

(B)

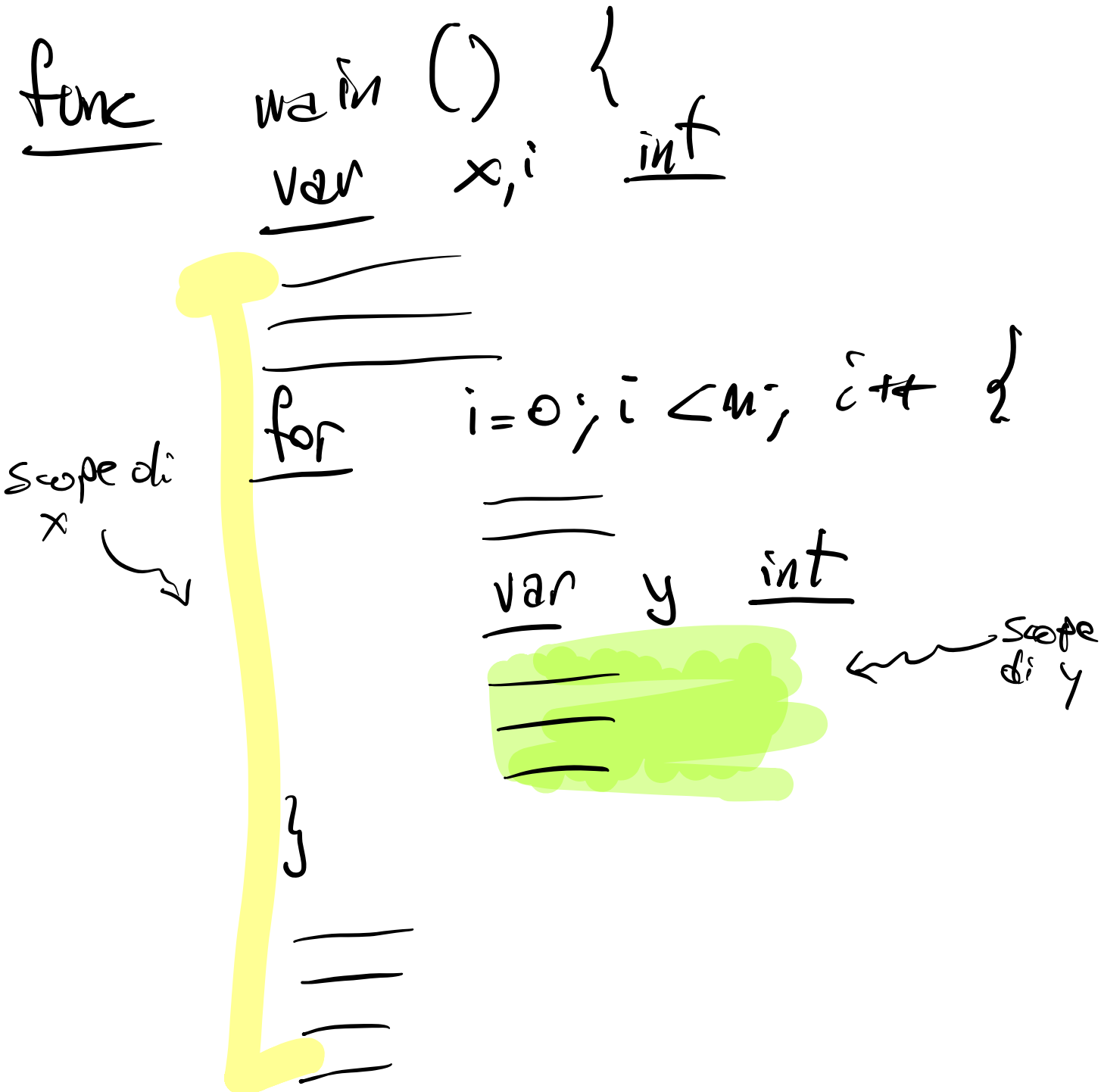
```
var x int
var y int
var f float64
```

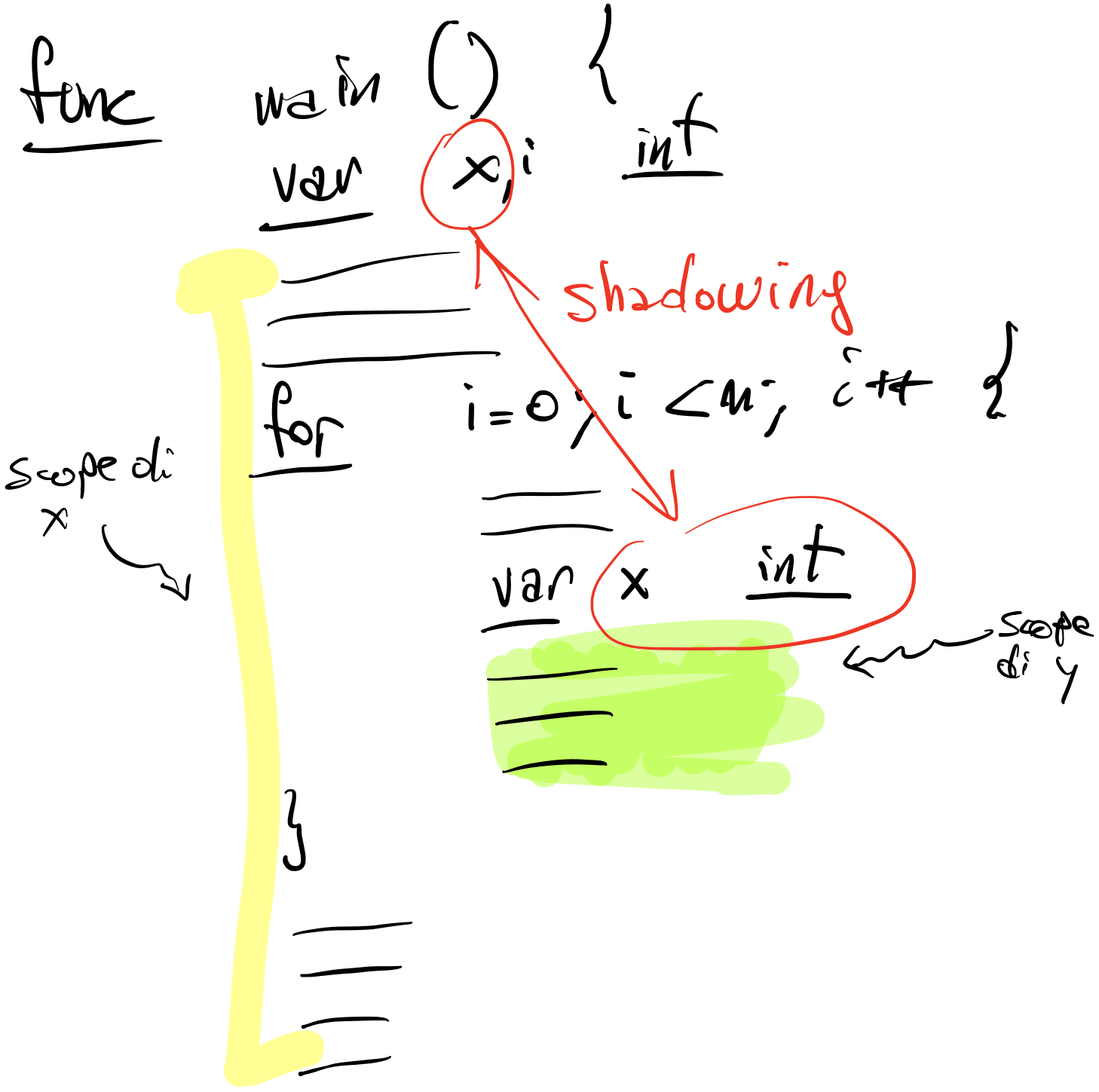
(C)

```
var (
    x, y int
    f    float64
)
```

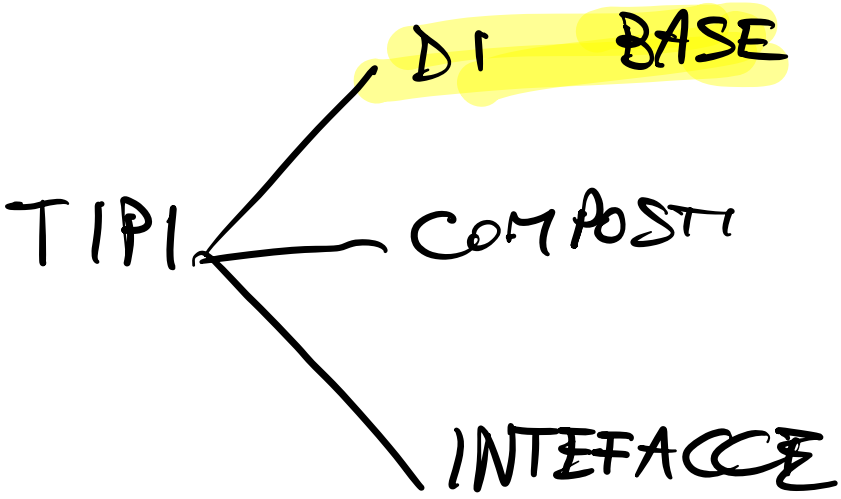
DICHIARAZIONE PRIMA DELL'USO

```
func main () {
    ____
    ____
    ____
    var x int
    ____
    ____
    ____
    ____
}
```


func main () {
var x, i int
for i = 0; i < n; i++ {
var y int
scope di x
scope di y
}


func main () {
var x, i int
shadowing
for i = 0; i < n; i++ {
scope di x
var x int
scope di y
}

TIPI IN Go



int

var x, y int

VALORE ZERO DI int è 0

a.go

```
var x, y int
fmt.Println(x, y)
```

```
$> go run a.go
0 0
$>
```

b.go

```
var x, y int
fmt.Print(x, y)
```

```
$> go run b.go
00 $>
```

OUTPUT

```
fmt.Print(_, _, _)
fmt.Println(_, _, _)
```

INPUT

```
fmt.Scan(&x)
```

ASSEGNAMENTO

variabile = espressione

```
var x, y int
x = 3
y = x
y = (x+1) * 4
x = (x+1) * (x+1)
```

```
var  x, y, z int
x = 5
→ y = (x+1) + z*3
x = y * x
z = x + y
fmt.Println(x, z)
```

x	30
y	6
z	36

30 36

1 CFU = 25 h LAVORO

12 CFU = 300 h LAVORO

72
50

122

ESPRESSIONI int

- COSTANTI int

  37, -2, -15, 1257

- VARIABILI int

- OPERATORI

  \+

  \-

  *

  / ← divisione intera

% ← resto della divisione

- PARENTESI ( )

3 * 7 + 4

4 + 3 * 7

(4 + 3) * 7

((4 + 3) * 7) / 3

((4 + 3) * 7) % 3

```
var x,y int
fmt.Scan(&x)
y = x%10
fmt.Println(y)
```

Stampa unità

142 → 2

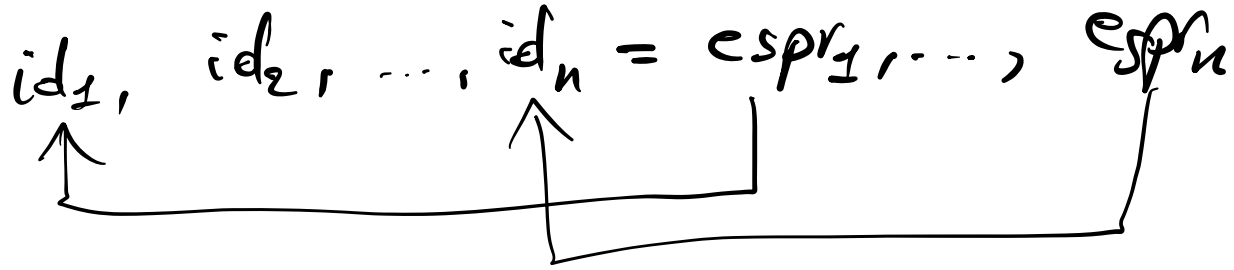
```
var x,y int
fmt.Scan(&x)
y = (x/10)%10
fmt.Println(y)
```

Stampa decine

142 → 4

(x%100)/10

ASSEGNAMENTI MULTIPLI



$$x, y = y, x$$

SHORT ASSIGNMENT

variabile := espressione

dichiarare + assegnare una variabile

var x int

for - - - {

x := 7

:

}