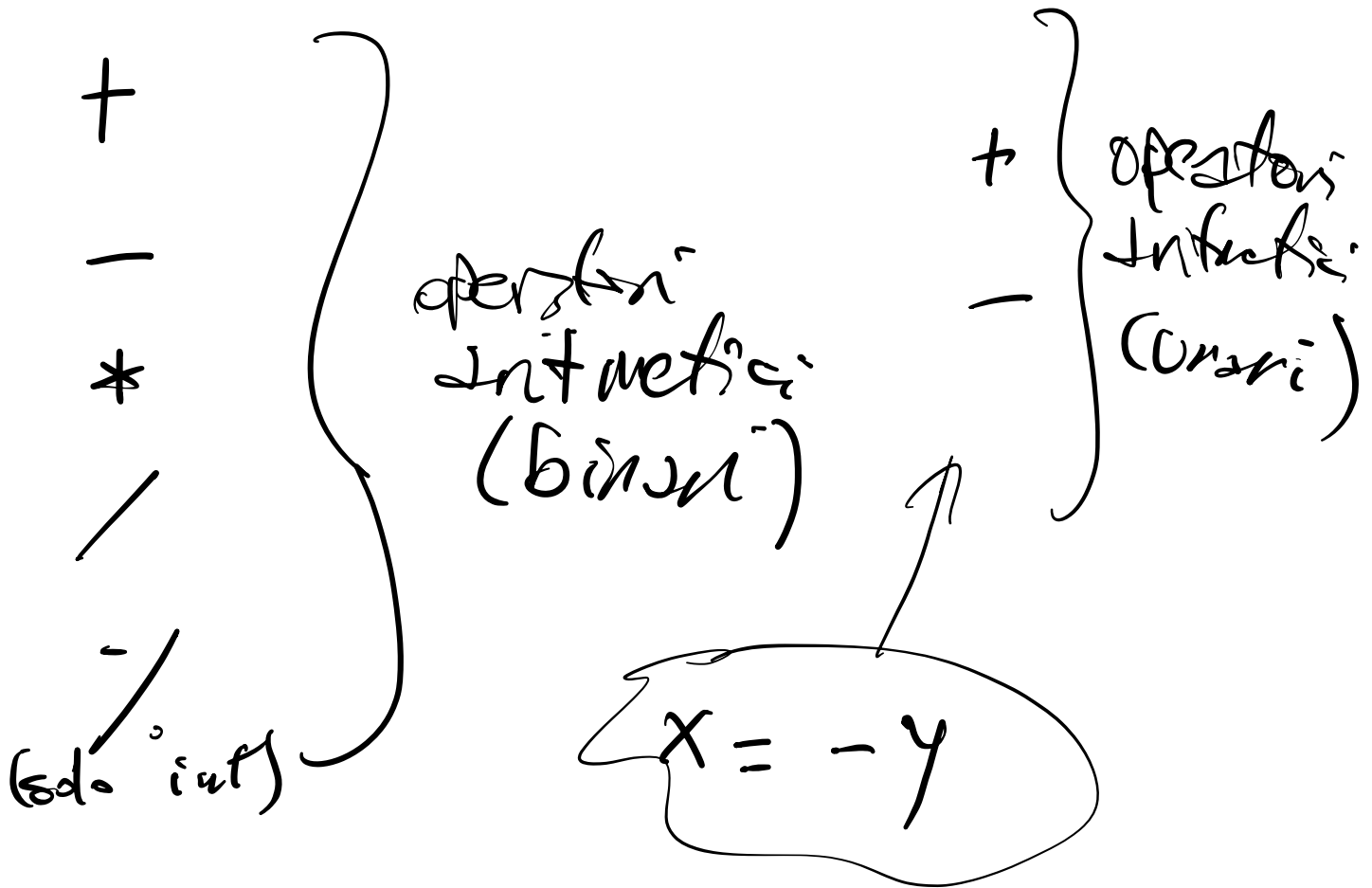


# OPERATORI int E float



## ISTRUZIONI INCREMENTO/DECR.

$x++$                        $x = x + 1$   
 $x--$

## OPERATORI DI ASSEGNAMENTO

$t =$                        $x = x + \text{ESPR.}$   
 $--$                        $//$

\* =

/ =

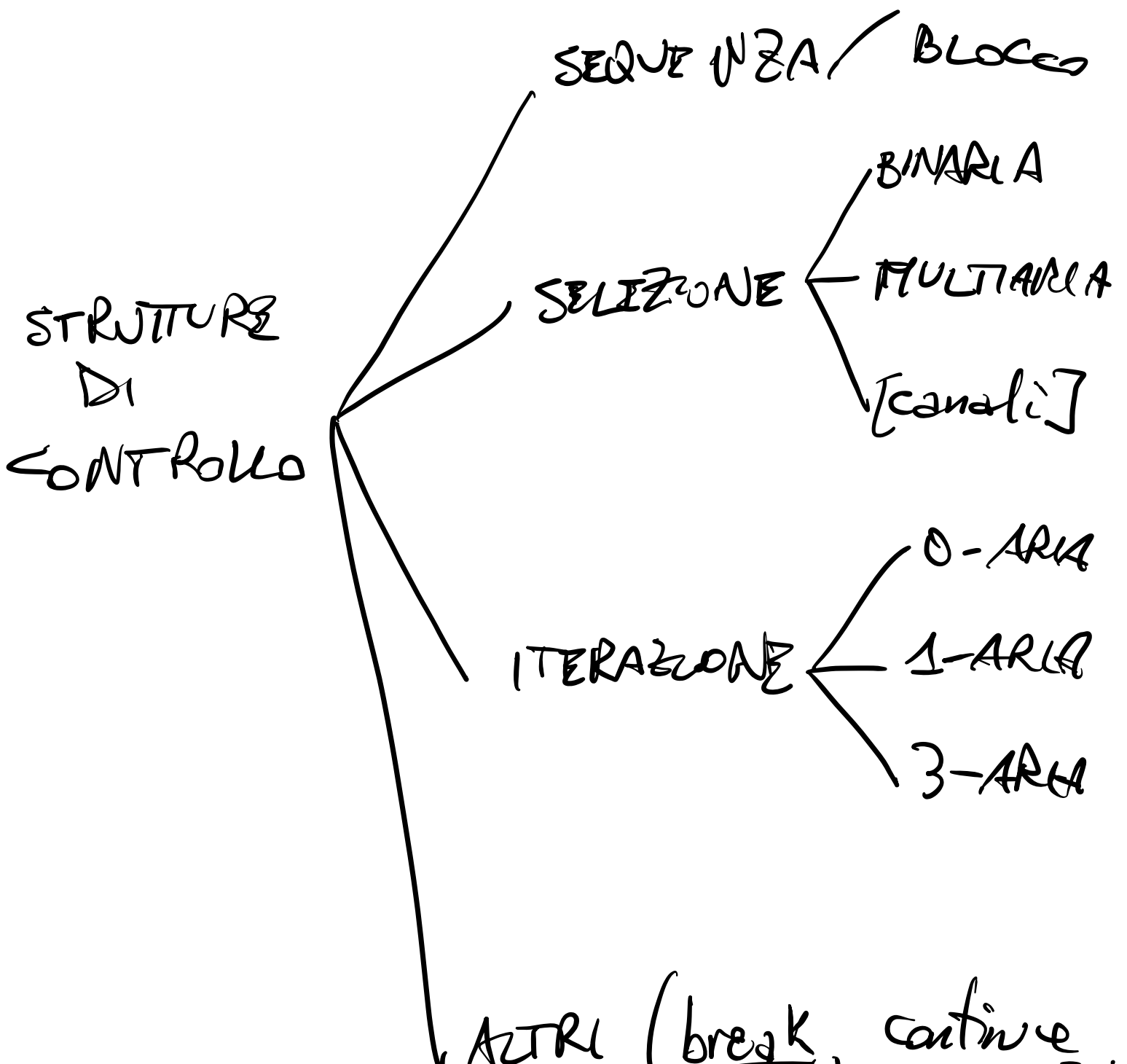
% = <sup>solo</sup>  
(int)

X+ = ΣESP

X = X + 1  
X++  
X += 1

# STRUTTURE DI CONTROLLO DEL FLUSSO

GOTO ← spaghetti coding

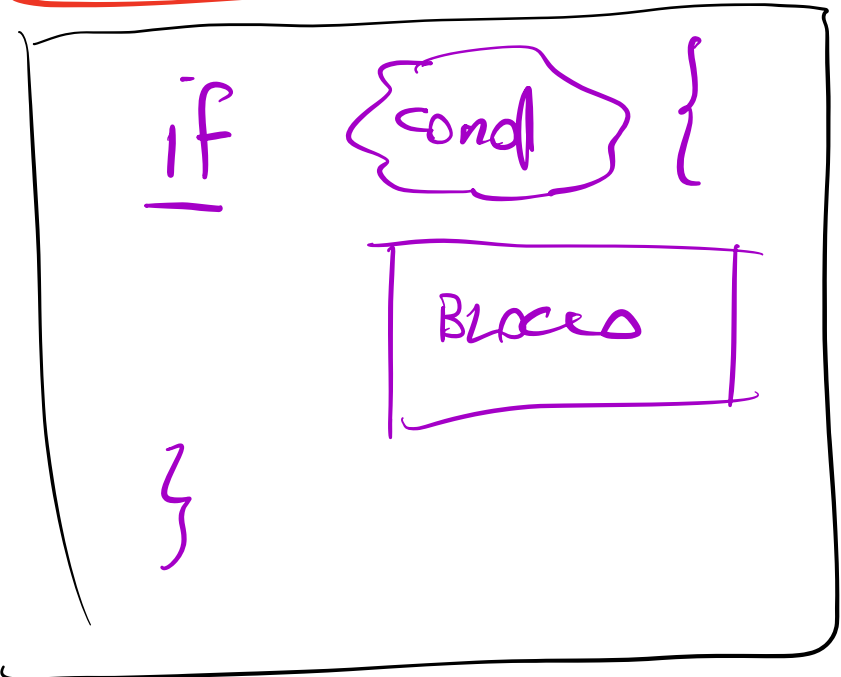


return, defer, ...)

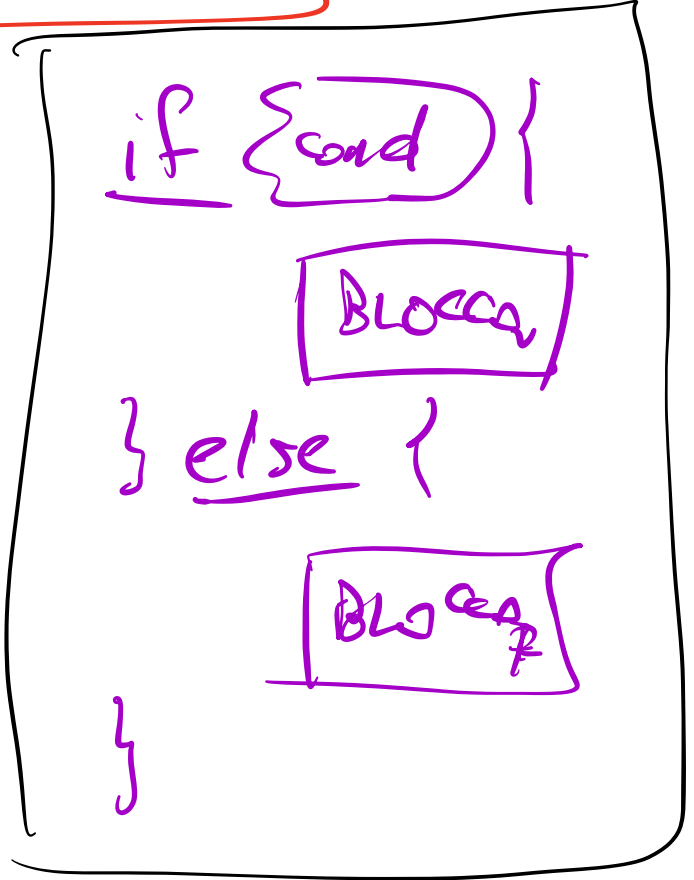
# SEQUENZA (BLOCCO)

{ ... }

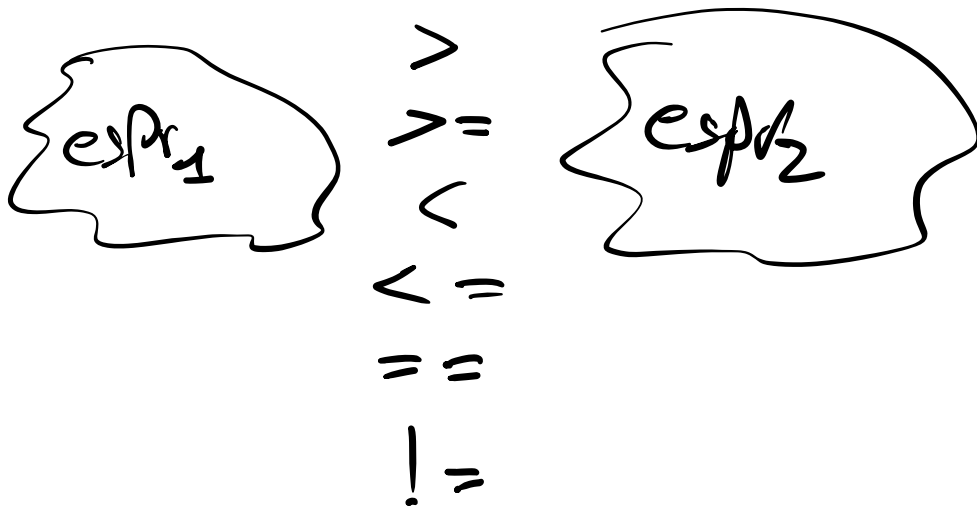
# SELEZIONE BINARIA



IF "incompleto"



IF "completo"



## ESERCIZIO

leggi due frazioni positive e  
 stabilisci se la prima  
 è minore della seconda.

var n1, n2, d1, d2    int

fut. Scan (&n1)

fut. Scan (&d1)

fut. Scan (&n2)

fut. Scan (&d2)

if  $n1 * d2 < n2 * d1$

fun. Println ("La prima è minore")

⊕

} else {

fun. Println ("La seconda è minore  
o uguale")

}

$$\frac{m_1}{d_1} < \frac{m_2}{d_2}$$

$$m_1 d_2 < d_1 m_2$$

⊕

fun. Print ("La prima è ")

if  $m_1 d_2 < d_1 m_2$  {

fun. Print ("minore")

} else {

fat. Print ("maggiore o uguale")  
}  
fat. Print ("della seconda")

## ESERCIZI

- Date due date, stabilire se la prima precede la seconda
- Dati  $a, b, c$  decidere se l'equazione

$$2x^2 + bx + c = 0$$

- ha due soluzioni reali distinte
- stabilire se la somma di due interi ha la cifra delle decine

uguale a +

```
if (Cond1) {
```

```
    Blocco1
```

```
} else if (Cond2) {
```

```
    Blocco2
```

```
} else if (Cond3) {
```

```
    Blocco3
```

```
} ... else {
```

```
    Blocco
```

```
}
```

if "a cascata"



# CONDIZIONI

espr<sub>1</sub>

>  
<  
>=  
<=  
==  
!=

espr<sub>2</sub>

operazioni  
relazionali

espressioni

bool

true

false

$\frac{\text{var } n1, d1, y}{\text{var } x} \quad \frac{\text{int}}{\text{bool}}$

$$x = (n1 * d1) > 4 * y$$

funct. Præntln (x)

OPERATOR	<u>bool</u>	(LOGIC)
----------	-------------	---------

&&	(AND)	} operatör binäri
	(OR)	
!	(NOT)	} operatör unäri

	Y	false	true
X	false	false	false
	true	false	true

x && y

"e"

CANGWINDONE  
LOGLOA

x \ y	false	true
false	false	true
true	true	true

x || y  
 "0"  
 DISGIUNZIONE  
 LOGICA

x	
false	true
true	false

! x  
 "not"  
 NEGAZIONE  
 LOGICA

---

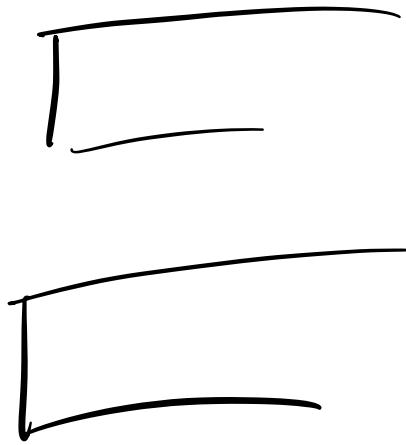
Var a, b    int  
Var x    bool

$$x = a > 0 \ || \ b > 0$$

if  $n1 < 0$  ||  $d1 < 0$  ||  $n2 < 0$  ||  $d2 < 0$  {  
    fmt.Println("ERROR")

} else {  
    if  $n1 * d2 < n2 * d1$  {

    } else {



    }

}

	f	t
f	f	f
t	f	f

	f	t
f	t	t
t	f	f

...

# IDENTITÀ BOOLEANE

$a \&\& a = a$  (idempotenza)

$!!a = a$  (doppio negare)

$a \&\& (a \parallel b) = a$  (assorbimento)

```
if x > 0 && (y != 0 || x > 0) {
```

...

```
}
//
```

if  $x > 0$  {  
...  
}

## Leggi di De Morgan

$$\neg (x \&\& y) = (\neg x) \vee (\neg y)$$

$$\neg (x \vee\vee y) = (\neg x) \&\& (\neg y)$$

$$\neg (x >> y) = x \&\& \neg y$$



$$\neg (x >> y) = x \&\& \neg y$$

if



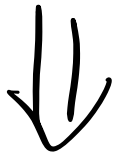
if  $x \leq 5$  &&  $y > x$

---

if !  $(x * x > 2 \text{ \&\& } (x > 0 \text{ || } y < 0))$



if  $x * x \leq 2 \text{ || } !(x > 0 \text{ || } y < 0)$



if  $x * x \leq 2 \text{ || } (x \leq 0 \text{ \&\& } y \geq 0)$

PRECEDENZA

$$(x * y) + 5$$

$$x + (y * 5)$$

$$(x + y) * 5$$

SHORT-CIRCUIT EVALUATION

if  $x > 5$  ||  $(x < 3 \ \&\& \ y + 5 > 3)$

if  $x == 0$  ||  $y/x > 5$