



# Fondamenti di Informatica

## Ingegneria Clinica

Lezione 26/10/2009



**Prof. Raffaele Nicolussi**

**FUB - Fondazione Ugo Bordoni**

**Via B. Castiglione 59 - 00142 Roma**



<b>Docente</b>	<b>Raffaele Nicolussi</b>	<a href="mailto:rnicolussi@fub.it">rnicolussi@fub.it</a> <b>0654803323</b>
<b>Lezioni</b> Aula 54 (ex aula 4) Via del Castro Laurenziano, 7	<b>Lunedì, Giovedì, Venerdì</b>	<b>12:00 – 13:30</b>
<b>Esercitazioni</b> Aula 15 Via Tiburtina, 205	<b>Giovedì</b>	<b>14:00 – ???</b>
<b>Ricevimento:</b>	<b>Per appuntamento</b>	<b>in FUB, per email, per telefono</b>
<b>Sito web:</b>	<b><a href="http://w3.uniroma1.it/IngClinFondinf">http://w3.uniroma1.it/IngClinFondinf</a></b>	



## Operatori di uguaglianza e relazionali

- ❑ usati nei test
- ❑ gli operatori relazionali hanno stesso livello di priorità
- ❑ gli operatori di uguaglianza hanno un livello più basso
- ❑ entrambi hanno associatività da sinistra a destra

<b>Operatori di uguaglianza</b>		
Uguale a	<b>==</b>	<b>x == y</b>
Diverso da	<b>!=</b>	<b>x != y</b>
<b>Operatori relazionali</b>		
Maggiore	<b>&gt;</b>	<b>x &gt; y</b>
Minore	<b>&lt;</b>	<b>x &lt; y</b>
Maggiore o uguale	<b>&gt;=</b>	<b>x &gt; = y</b>
Minore o uguale	<b>&lt;=</b>	<b>x &lt; = y</b>

- ❑ danno risultati booleani, ossia **vero** o **falso**
- ❑ in C
  - **falso** (false) equivale a **zero**
  - **vero** (true) equivale a qualsiasi valore **diverso da zero** (come valore di un test viene usato **uno**)



## Priorità

Operatore	Associatività
( )	da sinistra a destra
* / %	da sinistra a destra
- +	da sinistra a destra
< <= > >=	da sinistra a destra
== !=	da sinistra a destra
=	da destra a sinistra

Espressione	Espressione equivalente	Valore
$j > m$	$j > m$	0
$m/n < x$	$(m/n) < x$	1
$j \leq m \geq n$	$(j \leq m) \geq n$	1
$j \leq x == m$	$(j \leq x) == m$	1
$-x + j == y > n \geq m$	$((-x) + j) == ((y > n) \geq m)$	0
$x = x + (y \geq n)$	$x = (x + (y \geq n))$	3.5
$j+1 == m \neq y*2$	$((j+1) == m) != (y*2)$	1

`int j=0, m=1, n=-1;`

`float x=2.5, y=0.0;`

Università degli Studi "La Sapienza" – Fondamenti di Informatica



```
#include <stdio.h>
int main (void)
{   int j=0, m=1, n=-1, tot;
    float x=2.5, y=0.0, ris;
    ris= x + (y>=n);
        printf("\nRisultato:\t%f\n", ris);
    tot=((j+1)==m)!= (y*2);
        printf("\nTotale:\t%d\n", tot);
        printf("Totale in float:\t%f\n", tot);
    ris=((j+1)==m)!= (y*2);
        printf("\nRisultato intero:\t%d\n", ris);
        printf("Risultato float:\t%f\n", ris);
    tot= ((-x)+j) == ((y>n)>=m);
        printf("\nTotale:\t%d\n", tot);
        printf("Totale in float:\t%f\n", tot);
return 1; }
```

## Quiz

Prova a calcolare da solo cosa stampano le varie **printf**

**Risultato: 3.500000**  
**Totale: 1**

**Totale in float: -0.000000**  
**Risultato intero: 0**  
**Risultato float: 1.000000**

**Totale:0**  
**Totale in float: 0.0000000**



## Spiegazione Quiz

- `int j=0, m=1, n=-1, tot;`
- `float x=2.5, y=0.0, ris;`
- `ris = x + (y >= n);`
  - $2.5 + (0.0 \geq -1)$
  - $2.5 + 1 = 3.5$
- `tot = ((j+1) == m) != (y*2);`
  - $((0+1) == 1) != (0.0 * 2)$
  - $(1 == 1) != (0)$
  - $1 != 0 \rightarrow 1$



## Spiegazione Quiz 2

- $\text{tot} = ((-x) + j) == ((y > n) >= m);$
- $((-2.5) + 0) == ((0.0 > -1) >= 1)$
  - $(-2.5) == (1 >= 1)$
  - $(-2.5) == 1 \rightarrow 0$



## Strutture di controllo

- ❑ flusso normale: **esecuzione sequenziale**
- ❑ **trasferimento di controllo**: ci sono istruzioni che permettono di specificare che l'istruzione da eseguire non è quella successiva
- ❑ **da evitare l'uso del goto** (rende i programmi illegibili):
  - il teorema di Böhm-Jacopini ha dimostrato che i programmi possono essere scritti senza usare il goto
  - i programmi possono essere scritti usando solo tre strutture di controllo:
    - **sequenza, selezione e iterazione**
- ❑ **Strutture di selezione**
  - **if**
  - **if/else**
  - **switch**
- ❑ **Strutture di iterazione**
  - **while**
  - **do/while**
  - **for**





## Teorema di Böhm-Jacopini

- Il **teorema di Böhm-Jacopini**, enunciato nel 1966 dagli informatici Corrado Böhm e Giuseppe Jacopini, afferma che qualunque algoritmo può essere implementato utilizzando tre sole strutture, la **sequenza**, la **selezione** ed il **ciclo**, da applicare ricorsivamente alla composizione di istruzioni elementari



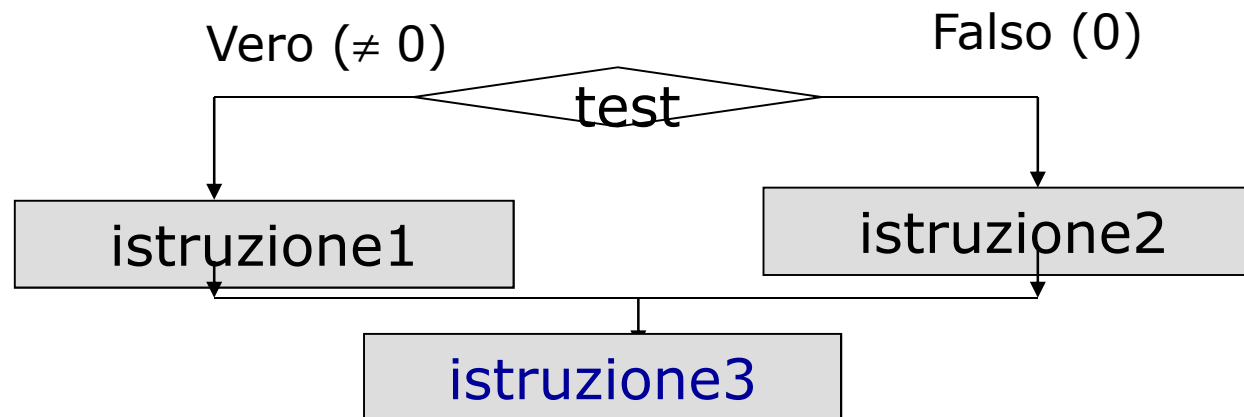
## Teorema di Böhm-Jacopini

- La **sequenza** è la normale elencazione di istruzioni perché vengano eseguite una di seguito all'altra nell'ordine in cui sono state scritte dal programmatore.
- La **selezione** è la scelta fra due percorsi da seguire successivamente, che dipende da una condizione che può essere vera o falsa.
- Il **ciclo**, detto anche iterazione, è un blocco di istruzioni che vengono ripetutamente eseguite fino a che una certa condizione cambia di stato.

# Strutture di selezione: if/else



```
if (<test>
    istruzione1;
else
    istruzione2;
istruzione3;
```

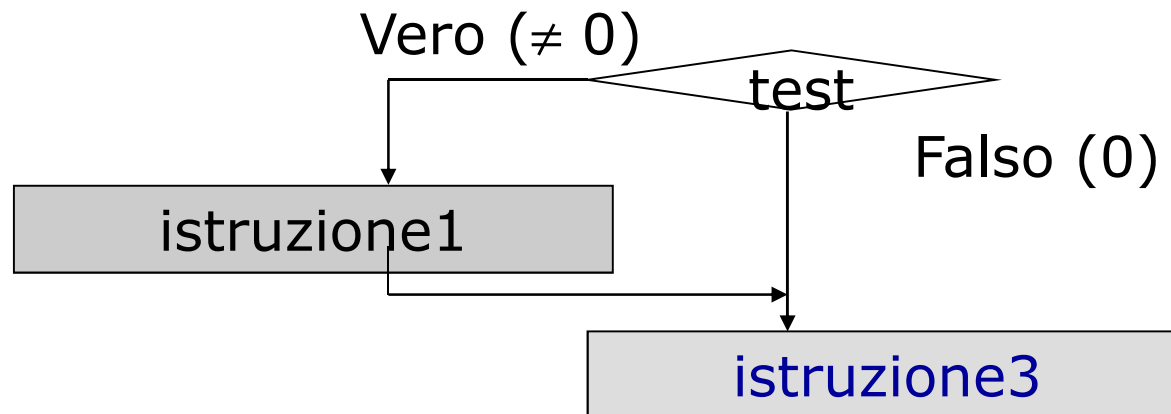


- ❑ Se il **<test>** è vero (diverso da zero), allora viene eseguita **istruzione1**;
- ❑ Se il **<test>** è falso (=0), allora viene eseguita **istruzione2**;



## Strutture di selezione: if, if/else

```
if (<test>)  
  istruzione1;  
istruzione3;
```



- ❑ Se il **<test>** è vero (diverso da zero), allora viene eseguita **istruzione1**;
- ❑ Se il **<test>** è falso (=0), allora viene eseguita **istruzione3**;

# Quiz





## Quiz :: regole del gioco

- Sono ammessi
  - Carta
  - Penna
  - Cervello
  - Una sbirciatina al compagno
- Non è ammessa
  - Confusione
  - Gente in piedi
- Tempo
  - 10 min ad esercizio



## Quiz ::

- ❑ Dato un intero, verifica se é minore di 100
- ❑ scrivere un programma per calcolare la radice quadrata di un numero
  - $a = \text{sqrt}(b)$  ;  $\rightarrow a = \sqrt{b}$
- ❑ Dati due interi che rappresentano il mese e l'anno, calcola il mese e l'anno successivi
- ❑ Problema: scrivere un programma per calcolare le radici di un'equazione di secondo grado

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$