

# Il Linguaggio C

Cap. 5 - Bellini Guidi

---

Prof. Salvatore Venticinque

Prof. Pietro Ferrara

# if-then-else

---

if (a > b)

Verifica Condizione

r = a + b;

Condizione Vera

else

r = a - b

Condizione Falsa

...

[Altre Istruzioni]

# Espressioni logiche

---

- Operatori di confronto

Simbolo	Significato	Utilizzo
<code>==</code>	uguale a	<code>a == b</code>
<code>!=</code>	diverso da	<code>a != b</code>
<code>&lt;</code>	minore	<code>a &lt; b</code>
<code>&gt;</code>	maggiore	<code>a &gt; b</code>
<code>&lt;=</code>	minore o uguale	<code>a &lt;= b</code>
<code>&gt;=</code>	maggiore o uguale	<code>a &gt;= b</code>

# Operatori logici

---

Simbolo	Significato	Utilizzo
&&	AND logico	a && b
	OR logico	a    b

<b>AND logico</b>	true	false
true	<b>true</b>	<b>false</b>
false	<b>false</b>	<b>false</b>

<b>OR logico</b>	true	false
true	<b>true</b>	<b>true</b>
false	<b>true</b>	<b>false</b>

# Teorema di De Morgan

---

- Il negato di un espressione booleana si ottiene:
  - negando tutti i termini
  - Sostituendo ogni operatore logico con il suo duale

Es:

`!((a > 0) &&(b >=0))`

`(a <=0) || (b<0))`

# If annidati

---

L'**else** si riferisce sempre all'ultimo **if** !!!!

```
if(temperatura < 20)
```

```
if(temperatura < 10) printf("Metti il  
cappotto!\n");
```

```
else printf(" Basta mettere una felpa");
```

# Parentesi e formattazione

---

```
if(temperatura < 20)
{
    if(temperatura < 10)
    {
        printf("Metti il cappotto!");
    }
    else
    {
        printf(" Basta mettere una felpa");
    }
}
```

# Solo formattazione

---

```
if(temperatura < 20)
```

```
    if(temperatura < 10) printf("Metti il  
cappotto!\n");
```

```
    else printf(" Basta mettere una  
felpa");
```



# Calcolare il massimo tra tre numeri

- Esercizio ...

# While

```
int main()
```

```
{
```

```
int a;
```

```
a = 5;
```

```
while (a>0)
```

```
  a = a - 1;
```

```
printf ("a=%d",a);
```

```
}
```

Condizione  
Vera



```
int main()
```

```
{
```

```
int a;
```

```
a = 5;
```

```
while (a>0)
```

```
{
```

```
  a = a - 1;
```

```
}
```

```
printf ("a=%d",a);
```

```
}
```

**Costrutto  
Iterativo**

# Algoritmo

---

Problema: Calcolo del Massimo Comun Divisore tra due numeri  $a, b$  :  $MCD(a, b)$

Soluzione di Euclide: “ogni divisore comune di  $a$  e  $b$  è divisore di  $a$ ,  $b$  e del resto  $r$  della divisione tra  $a$  e  $b$  ( $a \bmod b$ ), se questo non è nullo”

Algoritmo:

1. acquisire due numeri  $a, b$
2. se  $b > a$  scambiare  $a$  con  $b$
3. se  $b = 0$   $MCD(a, b) = a$  a andare al passo 6
4.  $r = a \bmod b$
5. sostituire  $a$  con  $b$ ,  $b$  con  $r$  ed andare al passo 3
6. Fine

# MCD Soluzione

---

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
int a=24;
```

```
int b=20;
```

```
int temp;
```

```
int mcd;
```

```
int r;
```

---

```
if (b>a) { //scambio di 2 variabili
    temp=a;
    a=b;
    b=temp;
}
while (b!=0) {
    r = a % b; //operazione modulo
    a=b;
    b=r;

    if (b==0)
        mcd=a;

} // end while
```

```
printf("mcd = %d",mcd);
}
```

# Precondizioni

---

Cosa succede se  $a$  o  $b$  sono uguali a  $0$  ??

Testiamo il programma!

# Input da tastiera

---

- Sintassi:

```
scanf(<formato>,<arg1>,...,<argn>);
```

< formato>: come per printf

< arg1>,...,<argn>: le variabili cui si vogliono assegnare valori

- **IMPORTANTE:**

I nomi delle variabili vanno precedute dall'operatore & che indica

l'indirizzo della variabile (vedremo più avanti il perchè)

- Esempio:

```
int x;
```

```
float z;
```

```
scanf("%d %f", &x, &z);
```

- 
- Istruzioni di input vanno viste come assegnazioni dinamiche:
    - L'assegnazione dei valori alle variabili avviene al tempo di esecuzione e viene deciso dall'utente
  - Assegnazioni tradizionali = Assegnazioni statiche
    - L'assegnazione dei valori alle variabili è scritta nel codice!



---

```
#include <stdio.h>
Int main()
{
    int a;
    float b;
    printf("Dammi un numero intero (A): ");
    scanf("%d",&a)
    printf("Dammi un numero reale (B): ");
    scanf("%f",&b) != 1)
}
printf("A=printf("B=%d\n",a);
%f\n",b);
```

# MCD con input da tastiera

---

```
int main()
{
    int a, b, temp;
    int mcd, r;

    printf("inserisci a: ");
    scanf("%d",&a);

    printf("inserisci b: ");
    scanf("%d",&b);
```

## Gestire preconditione e postcondizioni (1)

---

```
if ((a > 0) && (b > 0))
```

```
{
```

```
[...]
```

```
}
```

```
else
```

```
    printf("MCD definito solo per a > 0 e  
b > 0");
```

# Gestire preconditione e postcondizioni (2)

---

```
a = 0;
b = 0;
//while (!((a > 0)&&(b > 0)))
while ((a <= 0) || (b <=0))
{
    printf("inserisci a: ");
    scanf("%d",&a);

    printf("inserisci b: ");
    scanf("%d",&b);

    if((a <= 0) || (b <=0))
        printf("MCD definito solo per a > 0 e b >0, reinserire i valori\n");
}
```

# Il costrutto do-while

---

```
do {  
    Istruzione  
    Istruzione  
    Istruzione  
    Istruzione  
} while(condizione)
```

Costrutto di ciclo a condizione finale

Ripete almeno 1 volta il ciclo

## Gestire precondizione e postcondizioni (2)

---

```
do {  
    printf("inserisci a: ");  
    scanf("%d",&a);  
  
    printf("inserisci b: ");  
    scanf("%d",&b);  
  
    if((a <= 0) || (b <=0))  
        printf("MCD definito solo per a > 0 e b >0,  
reinscrivere i valori\n");  
}while ((a <= 0) || (b <=0));
```

# Esercizio

---

- Calcolo della media dei primi numeri diversi da 0

# Il costrutto FOR

---

```
for (int i=0; i< n; i++)
```

```
{
```

```
    Istruzione
```

```
    Istruzione
```

```
    Istruzione
```

```
}
```

*Esegue il ciclo esattamente **n** volte*





- Calcolare la media di  $n$  numeri, con  $n$  scelto dall'utente
- Calcolare il massimo di  $n$  numeri con  $n$  scelto dall'utente