

LIA - DEIS - Facoltà di Ingegneria - Università di Bologna  
**Elementi di informatica L**

Corso di laurea in Ingegneria Chimica - Anno Accademico 2007-2008

## Espressioni semplici e condizionali

---

### Esercizio 1

Progettare un algoritmo che dati tre numeri (voti)  $a$ ,  $b$ ,  $c$  che possono assumere i valori interi da 18 a 30, calcoli la loro media.

```
main()
{
    int a,b,c,media;
    a=18;
    b=19;
    c=22;
    media = (a+b+c)/3;
}
```

Provando poi ad eseguire il programma con il debugger (modificando il valore attribuito inizialmente alle variabili), si potevano fare le seguenti osservazioni:

- Le variabili vengono create all'inizio del blocco in cui sono definite. Vengono distrutte quando si esce dal blocco. Ad esempio, mettendo sotto osservazione col debugger la variabile `media`, Rhide potrebbe fornire i seguenti valori:
  - *not available* (non disponibile) finché il programma non viene eseguito
  - *13121* (valore casuale) quando si oltrepassa l'inizio blocco (la parentesi graffa)
  - *19* (risultato del calcolo) dopo aver superato l'assegnamento
  - *not available* (non disponibile) quando si esce dal blocco (si oltrepassa la parentesi graffa)
- Il calcolo viene effettuato con una divisione intera, quindi il risultato è approssimato. Anche definendo `media` di tipo `float` il calcolo viene effettuato con divisione intera. Infatti, l'operatore di divisione è sovraccaricato (overloaded): esso ha più significati (divisione intera, divisione fra reali, ...). Quindi viene selezionato l'operatore da usare in base *al tipo degli argomenti*, non di quello del risultato. Quindi, per ottenere una divisione reale, si devono inserire argomenti reali, ad esempio così:

```
float media = (a+b+c)/3.0;
```

---

### Esercizio 2

## Espressioni semplici

- Progettare e Codificare in C (scrivendolo in **voto1.c**) un programma che permetta il calcolo della media di tre voti dati in trentesimi. La media deve essere fornita in centesimi o trentesimi a seconda del valore assunto da una variabile *scala* di tipo carattere

```
main()
{
    int a,b,c,media;
    a=18; b=19; c=22;
    char scala='t';
    media = (scala=='t' ? (a+b+c)/3 : (a+b+c)/3*100/30);
}
```

**Nota:** Si sconsiglia la seguente soluzione (che pure in C funziona):

```
main()

{
    int a,b,c,media;
    a=18; b=19; c=22;
    char scala='t';
    (scala=='t' ? media=(a+b+c)/3 : media=(a+b+c)/3*100/30);
}
```

In questo caso, gli assegnamenti vengono effettuati come effetti collaterali all'interno della valutazione di un'espressione; il risultato dell'espressione viene poi scartato. Ovvero:  $(Cond ? RisTrue : RisFalse)$  è un operatore ternario (che ha 3 argomenti) e restituisce un valore. Se la condizione *Cond* è vera, restituisce il risultato dell'espressione *RisTrue*, altrimenti restituisce il risultato dell'espressione *RisFalse*. Quindi viene effettuato il confronto *scala=='t'*, se questo è vero viene valutata l'espressione  $media=(a+b+c)/3$ , che calcola la media dei 3 numeri, effettua come effetto collaterale l'assegnamento e restituisce il valore. Questo valore viene poi scartato. Come paragone, si pensi al seguente programma (corretto, in C):

```
main()
{int a,b;
(a=3) * (b=5);}
```

viene effettuato l'assegnamento *a=3*, che restituisce 3, poi viene effettuato *b=5*, che restituisce 5, poi viene effettuato il prodotto (15), che viene scartato.

---

### Esercizio 3

- Progettare e Codificare in C un programma che calcoli il valore assoluto di un numero *x*.

```
main()
{
    int x=10,val_ass;
    val_ass = (x>0 ? x : (-x));
}
```

---

### Esercizio 4

- Progettare e Codificare in C il programma che data una quantità di soldi in lire restituisca la

## Espressioni semplici

Progettare e codificare in C un programma che data una quantità di soldi in lire restituisca la quantità corrispondente in euro (/2000) o in dollari (/2500), a seconda del valore assunto da una variabile *moneta* di tipo char.

```
main()
{
    int lire=18000,conv;
    char moneta='e';
    conv = (moneta=='e' ? lire/2000 : lire/2500);
}
```

# Input / Output, Espressioni Condizionali e Cicli

---

## Esercizio 1

Si scriva un programma in linguaggio C, che legga da input le lunghezze dei lati di un triangolo e determini se il triangolo è equilatero, isoscele o scaleno.

```
#include <stdio.h>

int main(void)
{
    float primo, secondo, terzo;
    short uguali = 0; /* conta il numero di coppie di lati uguali */

    printf("Lunghezze lati triangolo ? ");
    scanf("%g%g%g", &primo, &secondo, &terzo);

    if (primo == secondo)
        uguali++;
    if (secondo == terzo)
        uguali++;
    if (primo == terzo)
        uguali++;
    if (uguali == 0)
        printf("Scaleno\n");
    else
        if (uguali == 1)
            printf("Isoscele\n");
        else
            printf("Equilatero\n");

    return 0;
}
```

**VARIANTE** che utilizza if-else annidati

```
#include <stdio.h>

int main(void)
{
```

## Espressioni semplici

```
float primo, secondo, terzo;

printf("Lunghezze lati triangolo ? ");
scanf("%g%g%g", &primo, &secondo, &terzo);

if (primo == secondo)
{
    if (secondo == terzo)
        printf("Equilatero\n");
    else
        printf("Isoscele\n");
}
else
{
    if (secondo == terzo)
        printf("Isoscele\n");
    else if (primo == terzo)
        printf("Isoscele\n");
    else
        printf("Scaleno\n");
}

return 0;
}
```

---

### Esercizio 3

Si scriva un programma in linguaggio C, che legga tre numeri interi positivi da input e ne determini il minimo, il massimo e la media.

```
#include <stdio.h>

main ( )
{
    int a, b, c;
    int min, max, somma;
    float media;

    printf("\nInserisci il primo numero: ");
    scanf ("%d", &a)
    min=a; max=a; somma=a; // oppure: min = max = somma = a;

    printf("\nInserisci il secondo numero: ");
    scanf ("%d", &b);

    if (b<min) min=b;
    if (b>max) max=b;
    somma=somma+b; // oppure: somma += b;

    printf("\nInserisci il terzo numero: ");
    scanf ("%d",&c);

    if (c<min) min=c;
    if (c>max) max=c;
    somma=somma+c ;

    media = (float) somma/3; // cast, perche' altrimenti divisione intera
```

## Espressioni semplici

```
/* VISUALIZZA I RISULTATI */

printf("\nIl valore minimo e': \t\t %d",min);
printf("\nIl valore massimo e': \t\t %d",max);
printf("\nLa media aritmetica e': \t %.2f",media);

}
```

---

### Esercizio 6

Si scriva un programma in linguaggio C, che:

1. legga un numero intero N;
2. legga N voti (compresi tra 18 e 30);
3. ne determini il massimo ed il minimo e la media.

```
#include <stdio.h>
main()
{
    int N,voto,max,min,i;
    printf("Inserire i voti e terminare con 0:\n");
    max = 18;
    min = 30;
    do {
        printf("Voto: ");
        scanf("%d",&voto);
        if (voto != 0) {
            if (voto>max) max = voto;
            if (voto<min) min = voto;
        }
    } while (voto != 0);
    printf("\nVoto max:%d\tVoto min:%d\n",max,min);
}
```

#### Alcune osservazioni:

- Poiché i voti sono compresi tra 18 e 30, i valori max e min sono stati inizializzati in maniera alternativa rispetto alla scelta precedente.
- E' necessario controllare che il voto inserito sia diverso da zero prima di aggiornare max e min.

#### Variante (con while):

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

## Espressioni semplici

```
main()
{
    int N,voto,max,min,i;
    printf("Inserire i voti e terminare con 0:\n");
    printf("Voto: ");
    scanf("%d",&voto);
    max = min = voto;
    while (voto != 0)
    {
        if (voto>max) max = voto;
        else if (voto<min) min = voto;
        printf("Voto: ");
        scanf("%d",&voto);
    }
    printf("\nVoto max:%d\tVoto min:%d\n",max,min);
}
```

**Osservazione:** è stato introdotto un ELSE per evitare di valutare la condizione del secondo IF; infatti, se  $voto > max$  allora sarà sicuramente  $voto > min$  e il secondo confronto può essere evitato. Si osservi che, con questa modifica, non sarebbe corretto (dal punto di vista logico) inizializzare MAX e MIN a 18 e 30 (perché?).