

3.10 Operazioni in virgola mobile

Le operazioni aritmetiche permesse sulle variabili `float` e `double` sono le stesse che per gli `int`, e si possono scrivere espressioni con variabili di tipo misto. In ogni espressione dove compaia una variabile `float` (`double`) il calcolo viene eseguito considerando le parti frazionarie in precisione semplice (doppia). Naturalmente quando si va ad assegnare il valore ottenuto a una variabile, se essa è di precisione inferiore al risultato può succedere che ciò che si ottiene non sia significativo. Per esempio, date le seguenti dichiarazioni

```
int i;  
float x;  
double y;
```

se alle richieste

```
printf("\n\n Digitare un valore reale: ");  
scanf("%f", &x);  
printf("\n\n Digitare un valore intero: ");  
scanf("%d", &i);
```

l'utente immette i valori 152.23 e 7, dopo gli assegnamenti

```
y = x;  
x = i+x;  
i = i+y;
```

la `printf`

```
printf("\n valore di x: %.2f  valore di i: %d", x, i);
```

visualizzerà i valori:

valore di x: 159.23 valore di i: 159

Ovviamente l'assegnamento di $i+x$ a i fa perdere al risultato la parte decimale. Ma se venissero immessi i valori 56489.45 e 7, la visualizzazione sarebbe (dipende comunque dall'implementazione C):

valore di x: 56496.45 valore di i: -9040

Quindi si utilizzino operazioni miste ma coscienziosamente.

Per ottenere la massima precisione possibile le funzioni matematiche che si trovano nella libreria `math.h` solitamente acquisiscono in ingresso e restituiscono valori `double`. Tra esse le trigonometriche

<code>sin(x)</code>	seno
<code>cos(x)</code>	coseno
<code>tan(x)</code>	tangente
<code>sinh(x)</code>	seno iperbolico
<code>cosh(x)</code>	coseno iperbolico

e altre di uso generale come

<code>log(x)</code>	logaritmo in base e di x
<code>log10(x)</code>	logaritmo in base 10 di x
<code>sqrt(x)</code>	radice quadrata

L'uso di variabili di alta precisione dovrebbe comunque essere limitato ai casi di effettiva utilità in quanto utilizzare un `float` al posto di un `int` o un `double` al posto di uno degli altri due tipi, oltre a portare a una maggiore occupazione di memoria, determina una maggior lavoro di elaborazione delle operazioni e quindi diminuisce i tempi di risposta. Inoltre, a nostro avviso, usare una variabile `float` invece di un `int` dove non sia necessario porta a una peggiore leggibilità dei programmi.