

## 3.9 Variabili di tipo virgola mobile

I numeri che hanno una parte frazionaria sono detti in virgola mobile (*floating point*). Per esempio:

152.23

-91.64

0.867

non possono essere memorizzati nelle variabili di tipo `int`. Le variabili che contengono tali valori sono di tipo `float`:

```
float x, y, z;
```

```
x = 152.23;  
y = 0.00008;  
z = 7E+20;
```

```
z = 7000000000000000000000000000;
```

Per visualizzare una variabile `float` all'interno del formato della `printf` si deve specificare dopo il solito simbolo `%` il carattere `f` se si desidera il numero decimale in virgola mobile, e il carattere `e` se lo si desidera in forma esponenziale (detta anche notazione scientifica). Come per i numeri interi, si può far seguire il simbolo di percentuale da un numero che specifica la lunghezza del campo in cui dovrà essere posto il valore opportunamente allineato a destra. Per esempio, `printf ("%15f", x)`, riserva quindici caratteri per la stampa di `152.23`, che verrà posto negli ultimi sette caratteri della maschera. La lunghezza del campo può essere seguita da un punto e dal numero di caratteri del campo che devono essere riservati alla parte decimale. Per esempio:

riserva 5 caratteri per la parte decimale dei 15 totali.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
							1	5	2	.	2	3	0	0	0

Le istruzioni:

restituiscono rispettivamente

visualizzano

```
152.23
8e-005
```

In memoria le variabili `float` vengono comunque rappresentate in una particolare notazione esponenziale, in modo da risparmiare spazio. Naturalmente i numeri reali sono infiniti mentre i sistemi di elaborazione devono fare i conti con le limitazioni fisiche proprie della macchina; dunque dobbiamo fare attenzione: stiamo lavorando con delle approssimazioni che calcoli successivi possono rendere inaccettabili. Per far in modo che la rappresentazione dei reali sia ulteriormente più precisa il C ha un altro tipo di dato in virgola mobile, detto `double`, che occupa uno spazio generalmente di 8 byte (64 bit) e che quindi permette di lavorare con numeri positivi e negativi nell'intervallo da  $1.7E-308$  a  $1.7E+308$ .

Il tipo `double` comunque non garantisce automaticamente di poter rappresentare un numero doppio di cifre significative, ma certamente migliora l'accuratezza delle operazioni aritmetiche e riduce l'effetto degli errori di arrotondamento ■.

Per poter visualizzare una variabile `double` con la `printf` si può utilizzare nel formato la solita notazione `%f` o la sequenza `%lf` (*long float*), con le stesse convenzioni viste per le variabili `float`. L'istruzione

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

memorizza il valore passato dall'utente nella variabile `float` `x`. Al posto di `%f` si può utilizzare indifferentemente `%e`. Analogamente per le variabili `double` si usa `%f` o `%lf` ■.

Ogni definizione di costante che includa un punto decimale fa sì che venga creata una costante di tipo `double`:

```
#define PI 3.14159
```

definisce la costante `PI` che può essere utilizzata all'interno del programma al posto del valore `3.14159`; naturalmente il valore di `PI` non può essere modificato ■.