

11.2 Tipi fondamentali

Il linguaggio C ha un insieme di tipi fondamentali associati alle principali unità di memoria di un calcolatore, e alle corrispondenti modalità d'uso più comuni. Così gli interi sono rappresentati dai tipi

```
char      int
```

mentre i numeri in virgola mobile sono rappresentati da

```
float     double
```

Il tipo `char` è un intero che generalmente ha una dimensione pari a un byte (8 bit). Il nome `char` deriva dal fatto che la dimensione deve essere sufficiente a contenere un carattere. Tipicamente, allora, una variabile di tipo `char` contiene valori positivi compresi tra 0 e 255. Questo però è solo l'uso più comune. Il C impone che un `char` erediti le operazioni tipiche di un intero, cioè somma, sottrazione, divisione, moltiplicazione e così via. Allora, anche se raramente usato, non è un errore assegnare un valore negativo a una variabile di tipo `char`:

```
char raro = -1;
```

Il tipo `int`, invece, è un intero la cui dimensione è di solito pari alla *parola* (*word*) della macchina; dunque nella maggioranza dei casi una variabile di tipo `int` occupa quattro byte (32 bit). La dimensione di una variabile di tipo `int`, comunque, cambia da macchina a macchina e l'unica relazione universale è:

```
sizeof(char) <= sizeof(int)
```

Ossia: la dimensione di un `char` è minore o uguale alla dimensione di un `int`.

Il tipo `float` corrisponde a un dato numerico di singola precisione in virgola mobile. Anche la dimensione di un `float` dipende dall'architettura della macchina in esame: spesso un `float` occupa 4 byte (32 bit).

Il tipo `double` corrisponde a un dato numerico in virgola mobile, ma in doppia precisione. Anche la dimensione del `double` dipende dalla macchina: in molti casi un `double` occupa 8 byte (64 bit). L'unica relazione universale è:

```
sizeof(float) <= sizeof(double)
```

I quattro tipi fondamentali possono essere ulteriormente specificati per mezzo di due tipi di qualificatori, i *qualificatori di dimensione*

```
short     e      long
```

e il *qualificatore aritmetico*

```
unsigned
```

I qualificatori di dimensione si applicano al tipo fondamentale `int`:

```
short int
```

definisce un dato numerico la cui dimensione è minore o uguale a quella di `int`:

```
sizeof(short int) <= sizeof(int)
```

mentre

```
long int
```

definisce un dato numerico la cui dimensione è maggiore o uguale a quella di `int`.

Il C standard ANSI prevede anche un tipo `long double`, che definisce un dato numerico la cui dimensione è maggiore o uguale a quella di `double`.

Riassumendo si ha:

```
sizeof(char) <= sizeof(short int) <= sizeof(int) <= sizeof(long int)
```

```
sizeof(float) <= sizeof(double) <= sizeof(long double)
```

Il qualificatore aritmetico si applica invece ai tipi `char` e `int`:

```
unsigned char
unsigned int
```

In entrambi i casi, con `unsigned` si fa uso dell'aritmetica senza segno. Questo significa che i valori numerici contenuti nelle variabili di tipo `unsigned char` e `unsigned int` sono sempre considerati come interi positivi. Nell'uso dei qualificatori la sintassi del C ammette delle abbreviazioni:

```
short          <---->  short int
long           <---->  long int
unsigned       <---->  unsigned int
unsigned short <---->  unsigned short int
unsigned long  <---->  unsigned long int
```

Il tipo `long double` non può invece essere abbreviato.

Abbiamo visto come la dimensione dei tipi dipenda dalle caratteristiche del processore della macchina ospite. Nella tabella seguente si riportano i dati che si ritrovano in alcune delle più diffuse architetture.

<u>Tipo</u>	<u>Dimensione</u>
<code>char</code>	1 byte
<code>short</code>	2 byte
<code>int</code>	4 byte
<code>long</code>	4 byte
<code>float</code>	4 byte
<code>double</code>	8 byte
<code>long double</code>	8 byte

Con il semplice programma illustrato nel Listato 11.1 si può verificare su qualunque macchina la dimensione dei tipi fondamentali.

```
#include <stdio.h>

main()
{
    int ch, in, sh, lo, fl, dd, ld;

    ch = sizeof(char);
    in = sizeof(int);
    sh = sizeof(short);
    lo = sizeof(long);
    fl = sizeof(float);
    dd = sizeof(double);
    ld = sizeof(long double);

    printf("La dimensione di un char è      %d\n", ch);
    printf("La dimensione di uno short è     %d\n", sh);
    printf("La dimensione di un int è          %d\n", in);
    printf("La dimensione di un long è           %d\n", lo);
    printf("La dimensione di un float è          %d\n", fl);
    printf("La dimensione di un double è          %d\n", dd);
    printf("La dimensione di un long double è %d\n", ld);
}
```

Listato 11.1 Visualizzazione della dimensione dei tipi fondamentali