

Variabili

Nicola Bicocchi

DIEF - UNIMORE

I tipi di dati numerici interi (IEEE 754-1985)

Nome	Dimensione	Descrizione
char	1 byte	intero con segno 8bit
unsigned char	1 byte	intero senza segno 8bit
short	2 byte	intero con segno 16bit
unsigned short	2 byte	intero senza segno 16bit
int	4 byte	intero con segno 32bit
unsigned int	4 byte	intero senza segno 32bit
long long	8 byte	intero con segno 64bit
unsigned long long	8 byte	intero senza segno 64bit

I tipi di dati numerici in virgola mobile (IEEE 754-1985)

Nome	Dimensione	Descrizione
float	4 byte	numero in virgola mobile a 32 bit
double	8 byte	numero in virgola mobile a 64 bit

Definizione di variabili

- Con definizione di variabile, si intende il modo con cui in un file che segue la sintassi del linguaggio C viene richiesto di riservare memoria per contenere un certo dato e gli viene assegnato un nome simbolico.
- La sintassi per definire una variabile in C è:

```
1 <tipo> <nome variabile> ;
```

- In C il punto e virgola viene utilizzato in diversi punti del linguaggio per indicare la fine della *cosa* che si sta facendo. In questo caso la fine della definizione.
- Il tipo può essere una delle parole riservate indicate precedentemente. Vedremo in seguito altri tipi più complessi.

Esempi di definizioni

```
1 char c;  
2 short s;  
3 int i;  
4 long long numero; float f;  
5 double radice;
```

- Deciso il tipo, bisogna scegliere un nome per fare riferimento alla variabile.
- *Quali sono i nomi validi?*

Identificatori

- In C gli identificatori utilizzabili per dare un *nome* a qualcosa possono contenere qualsiasi combinazione di:
 - lettere maiuscole e minuscole
 - numeri
 - il carattere underscore (`_`)
- L'unico ulteriore vincolo è che non possono cominciare con un numero.
- Le sequenze che iniziano con un numero sono costanti o letterali numeriche.
 - decimali: cominciano con una cifra da 1 a 9 e proseguono con altre cifre da 0 a 9.
 - ottali: cominciano con 0 e proseguono con altre cifre da 0 a 7.
 - esadecimali: cominciano con «0x» o «0X» e proseguono con altre cifre da 0 a 9 e con le lettere (maiuscole o minuscole) da «A» a «F».

Letterali numerici

- Col termine letterale si intende un valore costante del C.
- Letterali di tipo int:
 - Decimali: 123, 245681, ecc...
 - Ottali: 0123, 02456, ecc...
 - Esadecimali: 0x123, 0x245abc, ecc...
- Con il suffisso «u» è possibile specificare che il loro tipo è unsigned int, (123u unsigned int).
- Letterali di tipo double, definiti dalla presenza di un punto decimale:
 - 123. è un double, come anche 123.0 o 123.345
 - Notazione scientifica: 1.23e2, oppure 123.e-2 oppure 12e7
- Letterali di tipo float, definiti dalla presenza di un punto decimale e dal suffisso «f»

Sequenze di escape

Sequenza	Valore	Significato
\t	0x09	Tabulazione
\n	0x0A	A capo (LF)
\r	0x0D	Torna a inizio riga (CR)
\"	0x22	Doppie virgolette
\'	0x27	Singolo apice
\\	0x5C	Backslash
\num	-	Il numero num interpretato come ottale
\xnum	-	Il numero num interpretato come esadecimale

Inizializzazione di variabili

- Di default, le variabili non hanno un valore predefinito, quindi bisognerà successivamente eseguire comandi per assegnargli qualcosa.
- È possibile anche inizializzare una variabile, in modo che all'avvio del programma (o meglio all'inizio della loro vita) abbia un valore preciso:

```
1 <tipo> <nome-varibile> = <espressione> ;
```

Variabili

```
1  #include<stdio.h>
2
3  int main() {
4      int base = 5; int altezza = 4; int area;
5
6      area = base * altezza / 2;
7      printf("Area: %d", area);
8
9      return 0;
10 }
```

int base = 5; int altezza = 4; int area = 0;

- È una dichiarazione. base, altezza, area sono nomi di variabili. Le variabili rappresentano simbolicamente i dati all'interno dei programmi.
- Una variabile identifica una locazione (posizione) della memoria in cui può essere memorizzato un dato a cui il programma può accedere.

Proprietà delle Variabili

int area = 0;

- **Nome:** identifica la variabile. E' un identificatore C: sequenza di lettere, cifre, _ che non inizia con una cifra (es. a123b e _as_231 lo sono, 1ab no).
- **Tipo:** specifica il tipo del dato. Esempio: int area specifica il fatto che area rappresenta un valore intero.
- **Indirizzo:** della cella di memoria che contiene il dato. Se il dato occupa più celle, questo è memorizzato in celle consecutive e l'indirizzo è quello della prima cella.
- **Valore:** dato rappresentato dalla variabile in certo momento dell'esecuzione. Può cambiare (variabile) durante l'esecuzione.