

Variabili

Nicola Bicocchi

DIEF - UNIMORE

- Libro di testo “Programmare in C”
 - Cap. 5 (tutte le sezioni tranne 5.8, 5.9, 5.11)

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».

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> ;
```

Esempi di definizioni con inizializzazione

```
1 char c = 21;
2 short s = 0xffff;
3 int i = '7';
4 long long numero = 1234567890123;
5 float f = 3.141592f;
6 double radice = 1.4142135623730950488016887242097;
```

- In questo modo la memoria allocata per la variabile viene anche riempita con un valore iniziale durante la sua creazione (detta anche allocazione). Notate che per ora abbiamo utilizzato solo espressioni composte da letterali.

Variabili di sola lettura

```
1  const <tipo> <nome-varibile> = <espressione> ;  
2  const double pi_greco = 3.1415926535897932384;
```

- È possibile specificare che la variabile in memoria potrà essere solo letta, ovvero che è una variabile read-only, tramite la parola riservata *const*.
- Questa variabile è identica alle altre (ha un tipo, una locazione di memoria e viene inizializzata), se non per il fatto che nessun comando successivo potrà modificarne il valore.
- Tecnicamente la parola chiave *const* potrebbe anche essere posta dopo il tipo, ma è molto inusuale e sconsigliabile per gli studenti.

Definizione di più variabili nella stessa linea

```
1 <tipo> <nome-var-1>, <nome-var-2>, ... , <nome-var-N> ;  
2 int i, j, k;  
3 char a = 9, b = 23, c = 12;  
4 short s1, s2 = 33, s3;
```

- È possibile definire più variabili assieme per avere una scrittura più compatta. In pratica, si scrive il tipo una sola volta e si elencano di seguito le variabili che si intende definire.
- È inoltre possibile aggiungere o meno una inizializzazione ad ogni variabile.

Le variabili locali:

- sono variabili definite all'interno di una funzione (e.g., funziona main).
- sono allocate in una parte di memoria chiamata *stack*.
- il processo in esecuzione alloca automaticamente la memoria necessaria a contenerle nel momento in cui esegue la funzione.
- ogni variabile occupa uno spazio di memoria contiguo. le variabili, fra di loro, possono non essere contigue.

- Il tipo che associamo a una variabile identifica due caratteristiche principali:
 - la modalità di rappresentazione
 - la quantità di memoria occupata

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

Dimensione dei tipi

- La dimensione delle variabili numeriche intere possono variare a seconda delle architetture
- In alcuni contesti si può preferire utilizzare tipi di variabili non nativi, ma che ci danno indicazioni precise sulle dimensioni a prescindere dall'architettura impiegata (`int8_t`, `int16_t`, `uint8_t`, `uint16_t`, ..., definiti all'interno di `stdint.h`)

sizeof

```
1 printf("Size of:\n");
2 printf("\tchar=%lu\n", sizeof(char));
3 printf("\tshort=%lu\n", sizeof(short));
4 printf("\tint=%lu\n", sizeof(int));
5 printf("\tlong=%lu\n", sizeof(long));
6 printf("\tlonglong=%lu\n", sizeof(long long));
```

- In base alla rappresentazione, l'implementazione delle operazioni cambia completamente (e anche la loro complessità e le performance)
 - Numeri interi
 - Con segno $[-2^{b-1} : +2^{b-1} - 1]$ $b = 4$ $[-8, 7]$
 - Senza segno $[0 : +2^b - 1]$ $b = 4$ $[0, 15]$
 - Numeri virgola mobile

Range dei tipi di dato

- Variabili intere (limits.h) CHAR_MIN, CHAR_MAX, INT_MIN , INT_MAX, LONG_MIN, LONG_MAX , LLONG_MIN, LLONG_MAX, ...
- Variabili virgola mobile (float.h) FLT_MIN, FLT_MAX, DBL_MIN, DBL_MAX, ...

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 il 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

printf

...

variabili costanti

...