Variabili e tipi di dato in C

Alessandra Giordani
agiordani@disi.unitn.it
Lunedì 11 aprile 2011
http://disi.unitn.it/~agiordani/

Stringhe

- Un altro tipo di insieme che vorremmo poter rappresentare è quello delle stringhe di caratteri, cioè sequenze di caratteri, per poter comporre parole e frasi (ad esempio, per visualizzare messaggi sullo schermo).
- In C la rappresentazione delle stringhe è un argomento complesso. Per il momento limitiamoci ad considerare una stringa costante in C come una sequenza di caratteri compresi tra doppi apici ("). Ad esempio:
  - “ciao mondo!"
  - "questa è una stringa"
  - "per favore, inserisci un numero"

Stringhe (cont)

- Alcuni caratteri non sono rappresentabili con un singolo carattere (ad esempio, il carattere di fine riga che ci fa visualizzare un “a capo”)
- Delle sequenze speciali (chiamate escape sequences, sequenze di escape) ci consentono di inserire questi caratteri.
- Ad esempio, il carattere di fina riga puo essere rappresentato con '\n‘
- Quindi, la stringa "questa e una stringa\n" termina con un carattere di fine riga.

printf() – stringa costante

- Introduciamo le stringhe perchè ci servono per utilizzare la funzione printf che ci consente di stampare messaggi a video.

La funzione printf (che studieremo e definiremo più avanti) puo essere invocata in due modalità:

- printf(<una stringa costante>) stampa a video una stringa.
  - Ad esempio: `printf("Hello World!");`
  - visualizza sullo schermo il messaggio \Hello World!".

Variabili e valori

- Una variabile è un nome logico a cui è assegnato un valore.
- Quest'ultimo può cambiare nel corso dell'esecuzione di un programma.

```
…
x = 3
…
x = 7
```

Ciclo di vita di una variabile

- **Dichiarazione**: si informa il compilatore che esiste una certa variabile;
- **Inizializzazione/assegnamento**: si imposta il valore attuale della variabile (il termine “inizializzazione” si riferisce al primo assegnamento di una variabile);
- **Utilizzo**: si utilizza il valore associato per calcolare espressioni arbitrariamente complesse.

Il linguaggio C è

- *esplicitamente tipato*: occorre esplicitamente associare ad ogni variabile il tipo di dato che essa rappresenta
- *staticamente tipato*: il tipo di una variabile non puo cambiare durante il suo ciclo di vita;

Dichiarazione di una variabile

- Prima di poter essere utilizzata in qualsiasi altro modo, una variabile deve essere dichiarata associando al suo nome un tipo di dato.
- La più semplice istruzione di dichiarazione ha la forma:

```
<tipo di dato> <nome della variabile> ;
```

- dove il ; indica la fine dell'istruzione (tutte le istruzioni in C terminano con ;).

Nome delle variabili

- Il `<nome della variabile>` è un nome di comodo che scegliamo di associare ad una variabile.
- Il nome può avere qualsiasi lunghezza > 1
- Teniamo presente che il C è case sensitive, quindi i nomi `aBC` e `Abc` sono due nomi diversi!

Nome delle variabili

- Un nome può contenere solo:
  - lettere minuscole (a-z);
  - lettere maiuscole (A-Z);
  - cifre (0-9);
  - il carattere underscore (_).
  - Tutte le combinazioni di questi caratteri sono valide, tranne quelle in cui il primo carattere e una cifra

Nomi di variabili validi e non

- Nomi di variabili validi:
  - aBC
  - a
  - X
  - x2
  - ciao_sono_io
  - _234
- Mentre i seguenti non sono nomi validi:
  - 2aBC
  - 1
  - Ciao*()@

Nomi di variabili non validi

- Inoltre, non sono ammissibili come nomi di variabili tutti quei nomi che appartengono ad una lista di nomi riservati, già utilizzati dal C come costrutti sintattici o operatori:

auto	char	do	entry	for	const
if	register	sizeof	switch	unsigned	signed
break	continue	double	extern	goto	volatile
int	return	static	typedef	while	
case	default	else	float	enum	
long	short	struct	union	void	

Esempi di dichiarazioni

```
char a;        // dichiara una variabile di tipo “carattere” chiamata “a”
int b;         // dichiara una variabile di tipo “intero” chiamata “b”
int ciccio;    // dichiara una variabile “intera” chiamata “ciccio”
float decim; // dichiara una variabile “decimale” chiamata “decim”
double x;      // dichiara una variabile “decimale a doppia precisione”
               chiamata “x”
```

Non è possibile dichiarare più di una variabile con lo stesso nome!

Assegnamento di una variabile

- Dopo aver dichiarato una variabile è possibile assegnare un valore alla variabile.
- L'assegnamento di un valore ad una variabile ha la forma:

  <nome variabile> = <espressione>

- dove = è l'operatore di assegnamento del C (equivalente a ← in pseudocodice) e

  <espressione> può essere…

Esempi di assegnazione

```
int a, b;  // Si possono dichiarare piú variabili dello stesso tipo in una sola volta!
char c, d;
double e, f;
a = 7;
b = 10.7;  // Se assegniamo una costante decimale ad una variabile intera, la parte
           // decimale viene automaticamente troncata, cioé in questo caso nella
           // variabile immagazziniamo il valore 10!!
c = 'C';
d = ';';   // anche ';' é un carattere
e = 14567; // Se assegniamo un intero ad una variabile decimale, il numero viene
           // automaticamente convertito in intero
f = 456.789;
```

- In C il separatore delle cifre decimali è il punto (.) e non la virgola (,)
- Gli apici (‘) sono necessari per far capire al compilatore che intendiamo il carattere ‘a’ piuttosto che il contenuto della variabile a
- La possibilita di assegnare valori decimali ad una variabile dichiarata intera (e viceversa) e un'altra delle ragioni per cui il C può essere considerato un linguaggio debolmente tipato.

Esempi di assegnazione (cont)

```
int a, b;  // Si possono dichiarare piú variabili dello stesso tipo in una sola volta!
char c, d;
double e, f;
a = 7;
b = 10.7;  // Se assegniamo una costante decimale ad una variabile intera, la parte
           // decimale viene automaticamente troncata, cioé in questo caso nella
           // variabile immagazziniamo il valore 10!!
c = 'C';
d = ';';   // anche ';' é un carattere
e = 14567; // Se assegniamo un intero ad una variabile decimale, il numero viene
           // automaticamente convertito in intero
f = 456.789;
```

int somma;
float media;
somma=a;
somma=somma+b;
media=somma/2;

printf() - template

- printf può anche essere usata per stampare il contenuto di una variabile
- all'interno della stringa possiamo inserire delle sequenze di caratteri speciali a cui, in fase di stampa, l'interprete sostituirà il valore attuale di una variabile.
- In questo caso, dobbiamo anche dire alla funzione di quale variabile vogliamo stampare il valore:
  - per stampare una variabile int, la sequenza speciale è %d
  - per stampare una variabile float, la sequenza speciale è %f
  - per stampare una variabile char, la sequenza speciale è %c

printf() - template

- Queste sequenze possono apparire ovunque nella stringa (template) che passiamo a printf.
- Il valore della variabile verra stampato a video nella stessa posizione.
- Per dire alla funzione printf di quali variabili vogliamo stampare il valore, possiamo aggiungere alla funzione tanti argomenti quante sono le variabili di cui intendiamo stampare il valore.

printf() – esempio1.c

```
char a = 'B';
int x = 1000, y = 20000;
float k = 50.5;
printf("la variabile a vale: %c", a);
printf("la variabile x vale: %d", x);
printf("la variabile y vale: %d", y);
printf("la variabile k vale: %f", k);
printf("le variabili: k,x,a valgono rispettivamente:
                %f,%d,%c", k,x,a );
```

- riscrivere in un main vuoto e salvare come file *esempio1.c*, compilare ed eseguire per vedere cosa accade