

Linguaggio C: Variabili e Tipi di Dati

Variabili

- Con il linguaggio C è possibile associare un nome a quasi tutti gli oggetti: variabili, costanti, funzioni,...
- I nomi possono contenere:
 - lettere,
 - numeri,
 - carattere di sottolineatura “_” (underscore),
 - devono iniziare per lettera o con il carattere di sottolineatura.
- Un nome non deve coincidere con una parola riservata, né con il nome di una funzione di libreria, a meno che non si desideri realmente creare una propria versione della funzione di libreria

Variabili

- Un nome non deve coincidere con una parola riservata, né con il nome di una funzione di libreria, a meno che non si desideri realmente creare una propria versione della funzione di libreria
- Le parole riservate in C sono:

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

Variabili

- Tutte le variabili, prima di essere utilizzate, devono essere **dichiarate**, cioè va definita la associazione tra il *nome* della variabile e l'insieme dei *valori* che può assumere. Quella dichiarazione non è ritrattabile.
- L'insieme dei valori è definito come **tipo della dichiarazione o tipo del dato**.
- Variabili che possono assumere singoli valori sono chiamate **scalari**, mentre quelle che possono assumere coppie, terne o n valori sono chiamate **strutturate**.
- La dichiarazione generalmente viene fatta all'inizio del programma, ma in programmi di grandi dimensioni può trovarsi anche in altre posizioni (o altri file).

Variabili

- La dichiarazione di una variabile deve essere fatta **una** sola volta.
- Contemporaneamente o successivamente alla dichiarazione, la variabile può essere inizializzata (le viene assegnato un valore iniziale).

```
int x; //dichiarazione
x=0; //inizializzazione
int x=1; //dichiarazione e inizializzazione
int x,y,z;
```

Tipi di dati

I tipi di dati possono essere distinti in

- **built-in**, predefiniti nel linguaggio
 - **user-defined**, non definiti nel linguaggio, definiti dal programmatore sulla base di quelli predefiniti.
-
- Tipi built-in in C sono:

Tipi di dichiarazione	Rappresentazione	N. di byte occupati *
Char	Carattere	1 (8 bit)
Int	Numero intero	2 (16 bit)
Float	Numero reale	4 (32 bit)
Double	Numero reale in doppia precisione	8 (64 bit)

Tipo di dato **char**

- Utilizzato per dichiarare una variabile che potrà contenere un **solo carattere**.
- Generalmente occupa un byte
- La codifica utilizzata è l'ASCII.
- Saranno possibili operazioni di confronto (relazione d'ordine) tra variabili char in quanto, ad esempio, 'a' risulterà essere minore di 'b' poiché ad 'a'

corrisponde la rappresentazione decimale 97 e a 'b' 98.

`char x='b';` //dichiarazione ed inizializzazione con il carattere b

Tipo di dato **int**

- Il tipo INT permette di dichiarare variabili che potranno assumere
- valori interi o naturali.
- Il tipo `int` è per default *short*, cioè rappresentato su 2 byte
- Il tipo *long int* permette di estendere (utilizzando 2 byte in più) il
- range dei valori utilizzabili.
- Operatori aritmetici semplici per gli interi

	Simbolo	Esempio
Addizione	+	4+7=11
Sottrazione	-	5-9=-4
Moltiplicazione	*	3*7=21
Divisione Intera	/	10/3=3
Operatore modulo	%	10%3=1

Tipo di dato **float** e **double**

- I tipi FLOAT e DOUBLE vengono utilizzati per dichiarare variabili in **virgola mobile**. Essi corrisponderebbero ai numeri reali.
- La loro rappresentazione nella memoria utilizza un numero finito di bit, pertanto il numero di cifre dopo la virgola è necessariamente finito, così come l'insieme dei numeri rappresentabili è discreto.
- Float, 4 byte -> 6 cifre decimali dopo la virgola;
- Double, 8 byte -> 15 cifre decimali;
- Operatori applicabili sono operatori di confronto e operatori logici:

Uguaglianza	==
Diverso	!=
Minore	<
Maggiore	>
Minore uguale	<=
Maggiore uguale	>=

AND	&&
OR	
NOT	!

Tipo di dato e formato

- Le variabili possono essere usate non sono con gli operatori consentiti sul tipo di dato ma anche su altre istruzioni, ad esempio le funzioni di immissione **scanf()** e visualizzazione **printf()** Poichè queste possono essere usate con qualsiasi tipo di dato, è necessario denotare il tipo di dato specifico indicandone il **formato**

%d,%i	intero
%f	float
%g	double
%s	stringa
%c	carattere
%%	carattere %

Immissione/Visualizzazione dati

- Istruzione che consente di stampare a video, la sintassi è la seguente:

```
printf("stringa di formato",[lista parametri]);
```

- La stringa di formato può contenere un numero qualunque di specificatori di formato, ma il loro numero deve coincidere con il numero dei dati da stampare, passati come argomenti.

- E' possibile stampare il risultato di espressioni:

```
printf("Il risultato è: %d ",num1*num1);
```

- E' anche possibile introdurre, nella stringa di formato, delle specifiche di formattazione, `\n`, `\t`, `\b`

Immissione/Visualizzazione dati

- Istruzione che consente la lettura dallo standard input (tastiera) di caratteri o cifre che verranno memorizzate all'interno di opportune variabili:

```
scanf("stringa di formato",[lista parametri]);
```

- La stringa di formato contiene esclusivamente specificazione di conversione
- Nella lista dei parametri, a differenza della printf, la variabile deve essere preceduta dal simbolo **&**:

```
int i;
```

```
scanf("%d",&i);
```