

## ANSI C – Variabili e Costanti (Parte 2)

Eccoci alla seconda parte del tutorial dedicato alle variabili e alle costanti, riprendiamo il discorso da dove lo avevamo lasciato scrivendo finalmente qualche riga di codice.

```
/*
 * Inizio frammento di codice
 */
int primoAddendo      = 2;
int secondoAddendo    = 6;

int risultatoSomma = primoAddendo + secondoAddendo;
/*
 * Fine frammento di codice
 */
```

Ecco un breve frammento di codice che ci permette di vedere la dichiarazione, l'inizializzazione e l'utilizzo di variabili intere, ovvero che al loro interno contengono numeri naturali, sia positivi che negativi. Nella prima riga di codice andiamo a creare una variabile di tipo `int` e la inizializziamo a 2, successivamente creiamo un'altra variabile intera con valore 6 e in una terza variabile andiamo a mettere il risultato della somma delle altre due variabili. Da notare che la terza variabile viene inizializzata direttamente con il valore derivante dalla somma delle altre due variabili senza effettuare una preinizializzazione, che in questo caso sarebbe superflua. Notare che alla fine di ogni riga è presente il “;” non dimenticatelo altrimenti il compilatore vi segnalerà un errore.

Abbiamo visto quindi come dichiarare una variabile di tipo intero, ora vediamo un elenco degli altri tipi primitivi:

- `short` = Numero intero corto
- `long` = Numero intero lungo
- `float` = Numero reale (a virgola mobile)
- `double` = Numero reale a doppia precisione
- `char` = Carattere alfanumerico

Due parole sui tipi, i tipi `short`, `long`, si riferiscono a numeri interi, ovviamente quello che varia è la grandezza del numero che può essere inserito al loro interno, i tipi `float` e `double` sono chiamati numeri a virgola mobile, o più semplicemente sono i numeri reali, tra di loro differiscono per la rappresentazione ed in particolare per il numero di cifre dopo la virgola. Infine `char` è il tipo utilizzato per contenere i singoli caratteri, se si desidera scrivere una stringa di caratteri, bisogna usare gli array, ma di questo ne parleremo nei prossimi tutorial.

Veniamo ora alle costanti, quando si desidera assegnare un valore ad un nome in modo che questo non possa mai essere modificato all'interno del programma, si può procedere in due

modi, utilizzando una direttiva al compilatore, ovvero `#define`, oppure davanti alla dichiarazione della variabile inserendo la parola `const`, vediamo un esempio:

```
#define ALTEZZA 10

const int base = 20;

int area = (base * ALTEZZA) / 2;
```

Questo esempio è molto banale, che fa vedere l'utilizzo di entrambi i modi per definire una costante, veniamo ora alle differenze tra le due possibilità. Allora nel primo caso si tratta di una direttiva per il compilatore, in pratica ogni volta che, a tempo di compilazione, il compilatore incontra il nome che abbiamo deciso (nel nostro caso `ALTEZZA`), viene sostituito dal valore. Questa ovviamente si può inserire all'interno di un file `.h`, oppure all'inizio di un file, prima di andare a scrivere il codice sorgente. Il secondo caso invece andiamo a definire una variabile, in questo caso un numero intero, e lo andiamo a bloccare con `"const"`. Questo può essere definito in qualsiasi punto del programma, e non per forza all'inizio.

Per convenzione il nome della costante definita con il primo metodo si scrive sempre con il nome tutto maiuscolo.

Spero che anche questo tutorial sia stato chiaro e utile. Vi aspetto nel prossimo tutorial.

Manuel (aka Jekrom)  
per segnalazioni : [manuel.montini@gmail.com](mailto:manuel.montini@gmail.com)  
<http://iphonecodes.wordpress.com/>

#### LICENZA:

Questo tutorial è rilasciato sotto licenza Creative Commons : "Attribuzione-Non commerciale-Non opere derivate 2.5 Italia",  
<http://creativecommons.org/licenses/by-nc-nd/2.5/it/>