



Nuovi tipi di dato e costrutto condizionale

Alessandra Giordani

agiordani@disi.unitn.it

Lunedì 26 marzo 2012

<http://disi.unitn.it/~agiordani/>



Tipi di dato fondamentali

- **int** è il tipo di dato che consente di rappresentare numeri interi all'interno di un programma.
 - Es: 4
- **char** permette di rappresentare caratteri alfabetici.
 - Es: 'z'
- **float** permette di rappresentare numeri in virgola mobile
 - Es: 0.35
- **double** permette di rappresentare numeri decimali in virgola mobile in doppia precisione
 - Es: 3.1413



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!



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, cioè 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 possibilità di assegnare valori decimali ad una variabile dichiarata intera (e viceversa) è un'altra delle ragioni per cui il C può essere considerato un linguaggio debolmente tipato.



Template per printf() e scanf()

printf()

- Nel **template** dobbiamo 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

scanf()

- %d legge un int
- %c legge un char
- %f legge un float
- %lf legge un double

Ripasso sulle funzioni e I/O

- Scrivo a video un messaggio per la richiesta di un numero
- Leggo il numero
- Invoco una funzione passando come parametro quel numero
- Scrivo a video il risultato ritornato

```
#include <stdio.h>
#include <math.h>
```

```
int main()
{
    float num;
    printf("Inserisci un numero: ");
    scanf("%f", &num);
    printf("Il quadrato di %d e' %d\n", num, sqrt(num));
    return 0;
}
```

E' la funzione che prende in ingresso in intero e restituisce la sua radice quadrata. Definita in math.h

Compilare usando gcc -lm radice.c



Il costrutto if

Sintassi

```
if (<condizione>) { /* blocco_if ... */ }  
|[ else { /* blocco_else ... */ } ]
```

- Esegui `blocco_if` solo se `<condizione> != 0`
- Se presente, esegui `blocco_else` solo se `<condizione> == 0`



L'espressione condizionale

- Se è un valore uguale a 0 è falsa
 - Eseguo i comandi dell'else
- Altrimenti se è vera (valore diverso da zero)
 - Eseguo i comandi dell'if
- OPERATORI DI CONFRONTO
 - > maggiore
 - < minore
 - != disuguaglianza
 - == uguaglianza
 - Da non confondere con = dell'assegnamento

Radice di un numero negativo?!

- Controllo che il numero inserito non sia negativo.
- Se è negativo stampo errore
- Altrimenti stampo risultato dell'invocazione di sqrt

```
#include <stdio.h>
#include <math.h>

int main()
{
    float num;
    printf("Inserisci un numero: ");
    scanf("%f", &num);
    if (num<0)
        printf("Errore nel dato di ingresso: il numero è negativo.\n");
    else
        printf("Il quadrato di %d e' %d\n", num, sqrt(num));
    return 0;
}
```

Equivalentemente..

- Controllo che il numero inserito sia positivo.
- Se è positivo stampo risultato dell'invocazione di sqrt
- Altrimenti stampo errore

```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main()
```

```
{
```

```
    float num;
```

```
    printf("Inserisci un numero: ");
```

```
    scanf("%f", &num);
```

```
    if (num >= 0) ←
```

```
        printf("Il quadrato di %d e' %d\n", num, sqrt(num));
```

```
    else
```

```
        printf("Errore nel dato di ingresso: il numero è negativo.\n");
```

```
    return 0;
```

```
}
```

scanf() - matching multiplo - e if a cascata

- La funzione scanf puo' essere usata per leggere più di una variabile in un colpo solo:

```
#include <stdio.h>
int main()
{
    float peso, altezza;
    printf("Immetti il tuo peso in kg e la tua altezza in metri:\n");
    scanf("%f %f", &peso, &altezza);
    float imc=peso/(altezza*altezza);    //indice di massa corporea
    printf("Il tuo IMC e' %f. ", imc);
    if (imc<18.5)
        printf("Sei sottopeso, mangia!\n");
    else if(imc<25)
        printf("Alla grande!\n");
    else if (imc<30)
        printf("Sei un po' in carne, datti una regolata!\n");
    else printf("A dieta, subito!\n");
    return 0;
}
```



Caratteri e interi – 1

- La maggior parte dei linguaggi distingue i caratteri dai dati numerici: 5 è un numero mentre 'A' è un carattere
- In C, la differenza tra carattere e numero è sfumata: il tipo di dati `char` è un valore intero rappresentato con un byte, che può essere utilizzato per memorizzare sia caratteri che interi
- Per esempio, dopo la dichiarazione

`char c;`

i seguenti assegnamenti sono corretti ed equivalenti:

`c='A';`

`c=65;`

In entrambi i casi, viene assegnato alla variabile `c` il valore 65, corrispondente al codice ASCII della lettera A

Tabella Ascii

Dec Hx Chr Dec Hx Chr

32 20 SPACE	48 30 0	64 40 @	80 50 P	96 60 `	112 70 p
33 21 !	49 31 1	65 41 A	81 51 Q	97 61 a	113 71 q
34 22 "	50 32 2	66 42 B	82 52 R	98 62 b	114 72 r
35 23 #	51 33 3	67 43 C	83 53 S	99 63 c	115 73 s
36 24 \$	52 34 4	68 44 D	84 54 T	100 64 d	116 74 t
37 25 %	53 35 5	69 45 E	85 55 U	101 65 e	117 75 u
38 26 &	54 36 6	70 46 F	86 56 V	102 66 f	118 76 v
39 27 '	55 37 7	71 47 G	87 57 W	103 67 g	119 77 w
40 28 (56 38 8	72 48 H	88 58 X	104 68 h	120 78 x
41 29)	57 39 9	73 49 I	89 59 Y	105 69 i	121 79 y
42 2A *	58 3A :	74 4A J	90 5A Z	106 6A j	122 7A z
43 2B +	59 3B ;	75 4B K	91 5B [107 6B k	123 7B {
44 2C ,	60 3C <	76 4C L	92 5C \	108 6C l	124 7C
45 2D -	61 3D =	77 4D M	93 5D]	109 6D m	125 7D }
46 2E .	62 3E >	78 4E N	94 5E ^	110 6E n	126 7E ~
47 2F /	63 3F ?	79 4F O	95 5F _	111 6F o	127 7F DEL

Nota: il valore numerico di una cifra può essere calcolato come differenza del suo codice ASCII rispetto al codice ASCII della cifra 0 (es. '5'-'0' = 53-48 = 5)

Caratteri e interi – 2

- Le costanti di tipo carattere sono racchiuse tra apici singoli
- **Esempio:** Leggere un carattere da terminale e visualizzarne il codice numerico

```
/* Stampa del codice numerico di un carattere */
#include<stdio.h>

int main()
{
    char ch;

    printf("Digitare un carattere: ");
    scanf("%c", &ch);
    printf("Il codice numerico corrispondente e' %d\n", ch);
    return 0;
}
```

Caratteri e interi – 3

- Dato che in C i caratteri sono trattati come interi, su di essi è possibile effettuare operazioni aritmetiche
 - `int j = 'A'+'B'`
 - `j` conterrà il valore 131, somma dei codici ASCII 65 e 66
- Scrivere una funzione che converte un carattere da maiuscolo a minuscolo

```
#include<stdio.h>

char to_lower(char ch)
{
    return ch+32;
}

int main()
{
    char ch;

    printf("Digitare un carattere MAIUSCOLO [A-Z]: ");
    scanf("%c", &ch);
    printf("Ecco lo stesso carattere minuscolo: %c\n", to_lower(ch));
    return 0;
}
```

Funziona per la
codifica ASCII

Problema 1: se inserisco 'a'?

- Cosa succede se l'utente digita in numero che non è un carattere maiuscolo?
- Se il carattere immesso non è maiuscolo lo lascio com'è

```
#include<stdio.h>
```

```
char to_lower(char ch)
```

```
{
```

```
    char res;
```

```
    if ((ch>=65) && (ch<=90))
```

```
        //if (ch>='A') && (ch<='Z')
```

```
            res = ch+32;
```

```
    else
```

```
        res = ch;
```

```
    return res;
```

```
}
```

```
int main()
```

```
{
```

```
    char ch;
```

```
    printf("Digitare un carattere: ");
```

```
    scanf("%c", &ch);
```

```
    printf("Ecco il carattere minuscolo: %c\n", to_lower(ch));
```

```
    return 0;
```

```
}
```





Problema 2: se inserisco '?' Oppure '7'

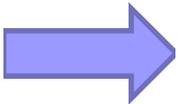
- Quando invoco la funzione *to_lower()* non controllo che il parametro sia davvero un carattere
- Utilizzo costruito condizionale. Condizione dell'if:
 - Valore del carattere compreso tra 65 e 90 oppure 97 e 122
 - Carattere nell'intervallo 'A' e 'Z' o tra 'a' e 'z'
 - Oppure, più semplicemente, utilizzo la funzione di libreria *isalpha(c)*
 - Restituisce un numero diverso da zero se l'argomento c è un carattere dell'alfabeto
 - Altrimenti restituisce zero

Funzione isalpha() - ctype.h

```
#include<stdio.h>
#include <ctype.h> /* incluso per la funzione isalpha */

char to_lower(char ch)
{
    char res;
    if ((ch>=65) &&(ch<=90))
        res = ch+32;
    else
        res = ch;
    return res;
}

int main()
{
    char ch;
    printf("Digitare un carattere alfabetico: ");
    scanf("%c", &ch);
    if (isalpha(ch))
        printf("Ecco il carattere minuscolo: %c\n", to_lower(ch));
    else
        printf("Hai inserito un carattere non alfabetico\n");
    return 0;
}
```





Cosa succede se...

- Se invece che invocare `isalpha(num)` richiamiamo la funzione `square(num)` o `sqrt(num)`?
- Il compilatore non da errori e produce l'eseguibile
- Ma come si comporta il nostro programma?
 - Il risultato è corretto in casi particolari
 - Il risultato è sempre sbagliato
 - Da errore



Attenzione: errori comuni

- Nella scanf() l'& ci va. Cosa succede altrimenti?
 - `int x;`
 - `scanf ("%d", x) ;`
- La parte sinistra di un assegnamento (L-value) rappresenta un luogo dove memorizzare un valore, non può essere un valore. Provate con...
 - `b-a = 100;`
- Nell'if ci va il doppio == (di confronto). Cosa ottenete con
 - `if (a=0)`
 - `if (b=1)`
- Il main è una funzione obbligatoria per ogni programma. Se non è presente una funzione main (cambiatele nome) che errore ricevete?