

Corso sul linguaggio C

Modulo 1

3 - Tipi di dato

M. Malatesta 3-Tipi di dato-10

1
09/03/2014

Prerequisiti

- Concetto di espressione
- Cenni sulla struttura del calcolatore

M. Malatesta 3-Tipi di dato-10

2
09/03/2014

Introduzione

In questa lezione si esaminano i **tipi di dato**.

I tipi di dato servono per poter distinguere ad esempio dati interi da dati di tipo reale e aiutano il programmatore a creare un programma che si adatti alla realtà del problema da risolvere.

M. Malatesta 3-Tipi di dato-10

3
09/03/2014

Informazioni generali

- Esistono diversi **tipi di dato** che possiamo utilizzare in base alle specifiche richieste del problema.
- Mediante questi tipi di dato possiamo **dichiarare costanti e variabili**

Le variabili possono essere manipolate con gli operatori come vedremo nella prossima Unità.

Vediamo in ordine questi concetti.

M. Malatesta 3-Tipi di dato-10

4
09/03/2014

Tipi di dato

I tipi di dato elementare che descriviamo sono:

- **Carattere**
- **Intero**
- **Float**
- **Double**

M. Malatesta 3-Tipi di dato-10

09/03/2014 5

Il tipo char

Il tipo di dato **char** si usa per rappresentare caratteri.

- Occupa **1 byte** di memoria.
- Con 1 byte è possibile rappresentare **256 diversi** possibili caratteri ($2^8=256$)
- La dichiarazione di una variabile di tipo char è:

char c;

- I caratteri costanti vengono indicati racchiudendoli tra apici ('). Ad esempio:

```
char car;           /* dichiarazione variabile di tipo carattere */
car='?';           /* assegnazione del carattere costante */
printf("%c", car); /* stampa a video il carattere '?' */
```

M. Malatesta 3-Tipi di dato-10

09/03/2014 6

Il tipo char

In programmazione i caratteri sono rappresentati mediante un codice detto **codice ASCII** (*American Standard Code For Information Interchange*) che associa a ciascun carattere un codice intero.

Per chiarire le idee, supponiamo di avere due funzioni **asc()** e **chr()** che rispettivamente trasformano caratteri in interi e viceversa.

asc: carattere → intero (codice ASCII)

chr: intero (codice ASCII) → carattere

Ad es. `asc('A')=65; chr(65)='A'; asc('B')=66; chr(66)='B';`

Il codice **ASCII** consta di 256 caratteri (da 0 a 255) ciascuno individuabile mediante il corrispondente valore intero.

Il tipo char

LETTURA

<code>scanf ("%c", &ch)</code>	Legge da input il carattere <i>ch</i>
<code>ch = getchar ()</code>	Legge un carattere come intero (codice ASCII)

SCRITTURA

<code>printf ("%c", ch)</code>	Scrive in output il carattere <i>ch</i>
<code>putchar (ch)</code>	Scrive in output il carattere di codice ASCII <i>ch</i>

Il tipo char

Esempi:

- 1) **char** ch;
 printf ("Immetti un carattere: ");
 scanf ("%c", &ch); /* legge un carattere */
- 2) **char** risposta;
 risposta = **getch**(); /* legge un carattere */
- 3) **char** ch = 'A';
 printf ("%c", ch); /* stampa il carattere 'A' */
- 4) **int** n = 65;
 putchar (n); /* stampa il carattere 'A' */

Il tipo char

Sul tipo **char** esistono delle **funzioni predefinite** che il programmatore può utilizzare per scopi particolari (v. esempio, a fine Unità)

if (isalpha (car)) istruzione	vero se <i>car</i> è alfabetico
if (isupper (car)) istruzione	vero se <i>car</i> è maiuscolo
if (islower (car)) istruzione	vero se <i>car</i> è minuscolo
<i>carl</i> = tolower (<i>car</i>);	converte in minuscolo
<i>carl</i> = toupper (<i>car</i>);	converte in maiuscolo
if (ispunct (car)) istruzione	vero se <i>car</i> è punteggiatura
if (isdigit (car)) istruzione	vero se <i>car</i> è una cifra
if (isspace (car)) istruzione	vero se <i>car</i> è lo spazio

Il tipo char

```
#include <stdio.h>
int main()
{
    char ch;
    puts("Immetti un carattere: ");
    ch=getchar();
    printf("Il suo codice ASCII è %d\n", ch);
    printf("Il carattere successivo e'");
    putchar (ch+1);
    getchar();
}
```

Il programma legge un carattere, ne stampa il codice ASCII e stampa il carattere successivo

La funzione **getchar()** legge il carattere *ch* come un intero (*casting*) .

Il carattere è stampato come intero (%d)

Si stampa il carattere successivo a *ch*

M. Malatesta 3-Tipi di dato-10

11
09/03/2014

Il tipo char

```
# include <ctype.h>
# include <stdio.h>
int main()
{ char car;
  printf("Immetti un carattere: ");
  car = getchar();
  printf("\nIl maiuscolo e' %c", toupper (car));
  getchar();
  return 0;
}
```

Il programma legge un carattere e lo trasforma in maiuscolo

La funzione **toupper** (*tipo*) restituisce il carattere maiuscolo. Fa parte delle funzioni sui caratteri e richiede la libreria **ctype.h**

M. Malatesta 3-Tipi di dato-10

12
09/03/2014

Il tipo int

Il tipo di dato **int** si usa per rappresentare variabili di tipo intero con segno.

- Occupa **2 byte** di memoria: 15 bit per il valore ed 1 bit per il segno
- Con 2 byte è possibile rappresentare valori che vanno da -65536 a $+65535$.
- In generale il range dei valori è individuato dalle due costanti **INT_MIN** e **INT_MAX**
- Per dichiarare una variabile int si scrive ad esempio:
`int conteggio; /* dichiarazione variabile conteggio */`

Il tipo float

Il tipo di dato **float** si usa per rappresentare variabili di tipo reale in virgola mobile.

- Il tipo **float** occupa **4 byte**.
- Con 4 byte è possibile rappresentare valori che vanno da $\pm 1.2E-38$ a $\pm 3.4E38$.
- Per dichiarare variabili di tipo float si scrive ad esempio:
- `float raggio; /* dichiarazione variabile reale raggio */`

Il tipo float

```
#include <stdio.h>
#include <math.h>
int main()
{
    float a,b,c;
    printf("Primo cateto: ");
    scanf("%f", &a);
    printf("Secondo cateto: ");
    scanf("%f", &b);
    c = sqrt (pow (a,2) + pow (b,2));
    printf("L'ipotenusa vale %4.2f", c);
    getchar();
    return 0;
}
```

M. Malatesta 3-Tipi di dato-10

Il programma legge le misure dei due cateti di un triangolo rettangolo e calcola l'ipotenusa

La libreria math.h serve per usare le funzioni
pow (b, e) – potenza di base *b* ed esponente *e*
sqrt (x) – calcola la radice quadrata di x

15
09/03/2014

Il tipo double

Il tipo di dato **double** si usa per rappresentare variabili di tipo reale in virgola mobile ma con una **precisione doppia rispetto al float**.

- Occupa **8 byte** di memoria:
- Con 8 byte è possibile rappresentare valori che vanno da $\pm 2.2E-308$ a $\pm 1.8E308$.
- Per dichiarare una variabile di tipo **double** si scrive ad esempio:
 - **double** raggio-orbita;

M. Malatesta 3-Tipi di dato-10

16
09/03/2014

Variabili

Per i nomi delle variabili si possono utilizzare:

- lettere maiuscole (A..Z)
- lettere minuscole (a..z)
- sottolineatura (_)
- cifre (0..9)

È buona abitudine:

- dare alle variabili nomi significativi
- scrivere le variabili in caratteri minuscoli

M. Malatesta 3-Tipi di dato-10

17
09/03/2014

Variabili

Esempi di dichiarazione di variabili:

```
int conteggio,  
    valore_iniziale;  
float somma1;
```

```
double asse_maggiore,  
        asse_minore;
```

N.B. – Il linguaggio C è *case-sensitive* ossia distingue i caratteri maiuscoli da quelli minuscoli. Quindi le variabili Asse, asse, ASSE, sono considerate diverse.

M. Malatesta 3-Tipi di dato-10

18
09/03/2014

Le variabili e il compilatore

Una delle maggiori conquiste delle tecnologie informatiche è la possibilità di lavorare con **linguaggi simbolici ad alto livello**.

Ciò significa che il programmatore può comodamente lavorare con **variabili mnemoniche**, invece che trattare con '0' ed '1'.

Sarà poi il **compilatore** a trasformare i nomi delle variabili nei rispettivi indirizzi di memoria **RAM**.

La tabella dei simboli

Il compilatore nella sua **tabella dei simboli** mantiene traccia della *corrispondenza tra i nomi delle variabili usate dal programmatore con i loro rispettivi indirizzi di memoria*.

conto	50100
valore	50102
totale	50104

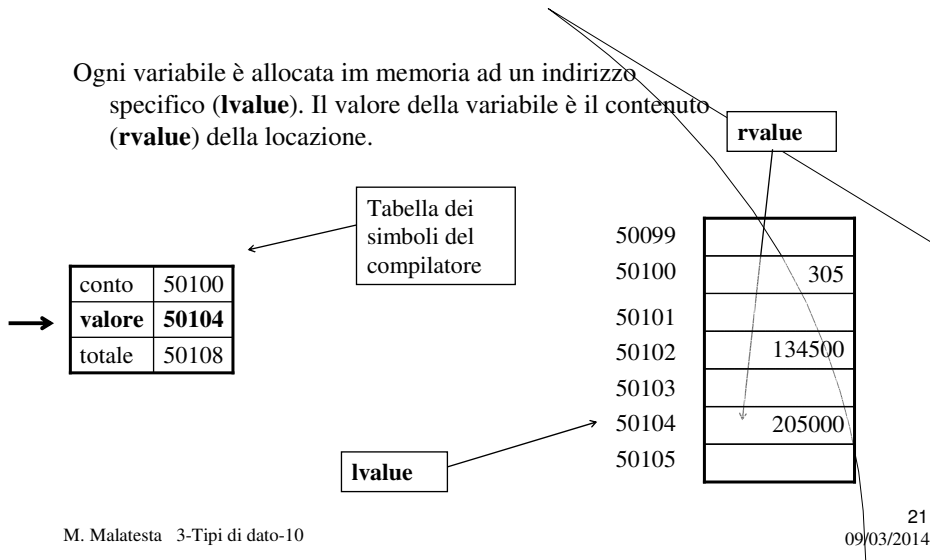
Tabella dei simboli del compilatore

In questo esempio supponiamo che un numero intero occupi 2 byte in memoria

50099	
50100	305
50101	
50102	134500
50103	
50104	205000
50105	

Ivalue e rvalue

Ogni variabile è allocata in memoria ad un indirizzo specifico (**lvalue**). Il valore della variabile è il contenuto (**rvalue**) della locazione.



M. Malatesta 3-Tipi di dato-10

Argomenti

- Tipi di dato
- Il tipo **char**
- Il tipo **int**
- Il tipo **float**
- Il tipo **double**
- Variabili
- Le variabili e il compilatore
- La tabella dei simboli
- Ivalue e rvalue

M. Malatesta 3-Tipi di dato-10

22
09/03/2014

Attività

1. Progettare un programma che calcoli il valore dell'espressione $a*x+b*y+c*z$, immettendo da input i valori a,b,c di tipo reale
2. I tre soci di un'azienda, A, B e C possiedono rispettivamente NA, NB e NC azioni. Il capitale iniziale dell'azienda è C1. A fine anno, l'azienda registra un capitale C2. Stampare il guadagno che ha conseguito ciascun azionista.
3. In un referendum si hanno i seguenti dati:
 - Numero degli iscritti a votare
 - Numero dei votanti
 - Numero dei SI'
 - Numero dei NO

Scrivere un programma che riproduca i risultati raccolti, stampando la percentuale dei votanti rispetto al totale degli iscritti e le percentuali dei SI' e dei NO rispetto al numero dei votanti.

M. Malatesta 3-Tipi di dato-10

23
09/03/2014

Altre fonti di informazione

- J. Purdum, C – ed. Jackson
- Romagnoli-Ventura, C/C++ - Ed. Petrini
- A.Bellini-A.Guidi, Conoscere il C – ed. McGraw Hill

M. Malatesta 3-Tipi di dato-10

24
09/03/2014