

Corso sul linguaggio C

Modulo 1

4 – Operare con i dati

M. Malatesta 4-Operare con i dati-02

1
09/03/2014

Prerequisiti

- Concetto di espressione
- Concetto di insieme e sue operazioni

M. Malatesta 4-Operare con i dati-02

2
09/03/2014

Introduzione

Combinando costanti e variabili, appartenenti ad un determinato tipo di dato T (**int**, **float**, ...), mediante opportuni **operatori** (*ammessi per il tipo T*) possiamo costruire **espressioni** di tipo T che saranno poi elaborate dal programma.

Vediamo in ordine questi concetti.

Operatori

Sulle variabili di ogni tipo di dato (**char**, **int**, **float**, **double**, ecc) possiamo effettuare delle elaborazioni mediante gli **operatori**, scrivendo delle **espressioni**.

Ogni operatore è caratterizzato da:

- un **simbolo** o un nome (ad es. +, -, *, /)
- l'**operazione** svolta (addizione, sottrazione, ..)
- determinate **regole di precedenza** tra operatori che servono a stabilire le priorità nel calcolo delle espressioni (che vedremo fra breve)

Operatori

Gli operatori che studiamo sono i seguenti:

- Assegnamento
- Operatori aritmetici
 - Operatori interi
 - Operatori reali
- Operatori relazionali
- Operatori logici
- Operatori sul bit

Assegnamento

L'operatore di **assegnamento**, che già conosciamo, è di fondamentale importanza. Esso consiste nell'attribuire ad una **variabile** un dato valore.

L'assegnamento si indica con il simbolo "=" e la sua sintassi è:

ident = espressione;

- dove *ident* è il nome che abbiamo scelto per la variabile ed *espressione* è una combinazione di operatori e variabili (tra breve ne daremo una definizione più precisa)

Assegnamento

Esempi:

- `c = 'A';` `/* assegna a c la costante 'A' */`
- `conteggio = 0;` `/* assegna il valore costante 0 */`
- `raggio = 0.5256;` `/* assegna il valore costante 5.256 */`
- `a = b = c;` `/* assegnazione multipla */`
- `Raggio = 5.256E-01;` `/* assegnazione di un valore reale */`

M. Malatesta 4-Operare con i dati-02

7
09/03/2014

Operatori aritmetici interi

Operatori **binari** (agiscono su due variabili) **interi**:

- Addizione (+)
- Sottrazione (-)
- Moltiplicazione (*)
- Divisione (/)
- Resto (%)

Operatori **unari** (agiscono su una sola variabile) **interi**:

- Autoincremento (++)
- Autodecremento (--)

M. Malatesta 4-Operare con i dati-02

8
09/03/2014

Operatori aritmetici interi

Ad esempio, date le variabili:

```
int a=3,b=2,c;
```

si ha:

Postincremento: prima assegna il valore e poi incrementa a di 1

Preincremento: prima incrementa di 1 il valore di a e poi lo assegna a c

Istruzione	a	b	c
c=a+b;	3	2	5
c=a-b;	3	2	1
c=a*b;	3	2	6
c=a/b;	3	2	1
c=a%b;	3	2	1
a++;	4		
a--;	3		
c=a+++;	4		3
c=++a;	4		4

a+++; equivale ad **a=a+1;**
a--; equivale ad **a=a-1;**

M. Malatesta 4-Operare con i dati-02

9
09/03/2014

Operatori aritmetici interi

In particolare per gli interi abbiamo gli operatori di **autoincremento** e **autodecremento**:

Autoincremento

- *Preincremento* (++) es: y=++x; incrementa x prima di assegnarlo ad y
- *Postincremento* (++) es: y=x++; incrementa x dopo l'assegnamento

Autodecremento

- *Predecremento* (--) es: y=--x; decrementa x prima dell'assegnazione
- *Postdecremento* (--) es: y=x--; decrementa x dopo l'assegnazione

M. Malatesta 4-Operare con i dati-02

10
09/03/2014

Operatori aritmetici reali

Questi operatori agiscono su variabili **reali**:

- Addizione (+)
- Sottrazione (-)
- Moltiplicazione (*)
- Divisione (/)
- Funzioni matematiche (presenti nella libreria **math.h**, che vedremo tra breve)
 - Ad es. **abs()**, **max()**, **min()**, **sin()**, **cos()**, **exp()**, **log()**, **pow()**, **rand()**, **sqrt()**, **tan()**.

Operatori relazionali

Questi operatori agiscono in genere su variabili **reali** o **intere**:

- Maggiore (>)
- Maggiore o uguale (>=)
- Minore (<)
- Minore o uguale (<=)
- Uguale (=)
- Diverso (!=)

Operatori logici

Gli operatori logici agiscono su variabili logiche. Una **variabile logica** è una variabile che può assumere soltanto 2 valori: **VERO** o **FALSO**

Un esempio di variabile logica è:

a = 5

Se il valore presente in a è 5 allora **a=5** ha valore **VERO**, altrimenti ha valore **FALSO**.

Esempi:

```
int flag1, flag2, uscita; /* dichiarazione variabili */
flag1 = 1; /* assegnazione costante logica 1 */
flag2 = flag1; /* assegnazione a flag2 del valore di flag1 */
uscita = (flag1 == 0); /* assegnazione ad uscita del valore di una
                        espressione logica */
```

M. Malatesta 4-Operare con i dati-02

13
09/03/2014

Operatori logici

Gli operatori logici sono:

- **AND (&&)** detto **prodotto logico**
- **OR (||)** detto **somma logica**
- **NOT (!)** detto **negazione**

In C i valori logici sono associati ai valori **0** e **1** nel seguente modo:

VERO	1
FALSO	0

M. Malatesta 4-Operare con i dati-02

14
09/03/2014

Tabella di verità

Il modo di operare degli operatori logici viene descritto mediante la loro **tabella di verità**.

Tabella di verità dell'operatore **AND**

&&	0	1
0	0	0
1	0	1

Tabella di verità dell'operatore **OR**

 	0	1
0	0	1
1	1	1

Tabella di verità dell'operatore **NOT**

!	
0	1
1	0

M. Malatesta 4-Operare con i dati-02

15
09/03/2014

Operatori sul bit

Esistono in particolare operatori in grado di lavorare **bit a bit** su un dato

&	And
 	Or
^	Or esclusivo
~	Not
<<	Shift a sinistra
>>	Shift a destra

M. Malatesta 4-Operare con i dati-02

16
09/03/2014

Operatori abbreviati

Per molti operatori esiste una forma abbreviata come mostrato in tabella

Forma abbreviata	La forma...	...sta per
<code>+= -= *= /= %=</code>	<code>x+=a;</code>	<code>x=x+a;</code>
<code>&= = ^=</code>	<code>x&=a;</code>	<code>x=x&a;</code>
<code><<= >>=</code>	<code>x<<=y;</code>	<code>x=x<<y</code>

M. Malatesta 4-Operare con i dati-02

17
09/03/2014

Definizione di espressione

Variabili e **costanti** possono essere legati tra loro mediante gli operatori del medesimo tipo per creare **espressioni** complesse

Possiamo scrivere espressioni per ogni tipo di dato, rispettando la sintassi degli operatori previsti.

Una espressione può essere:

- Una **costante**
- Una **variabile**
- Un **operatore** fra espressioni

M. Malatesta 4-Operare con i dati-02

18
09/03/2014

Esempi di espressioni

Espressione	Tipo espressione
5+3	Aritmetica Intera
b<c t!=r	Logica (vera se b<c o t diverso da r)
!y	Logica (vera se y = 0)
x==y && a>b	Logica (vera se x=y e a>b)
a+b%2	Aritmetica intera
5+2+(i<10)	Aritmetica intera (vale 8 se i<10, 7 altrimenti)
c	Variabile intera (se c è una variabile intera)
'a'	Costante di tipo carattere
"Rossi"	Costante di tipo stringa
rapporto	Variabile float (se rapporto è di tipo float)
0	Costante intera

M. Malatesta 4-Operare con i dati-02

19
09/03/2014

Precedenza degli operatori

Precedenza (*)	Simbolo	Operazione
1	!	Negazione
2	* / %	Moltiplicazione, divisione, resto
3	+ -	Addizione, sottrazione
4	> >= < <=	Maggiore, minore, ecc. ecc
5	== !=	Uguale, diverso
6	&&	And
7		Or
8	?:	Operatore ternario (**)
9	=	Assegnazione

(*) L'ordine di priorità può essere alterato mediante l'uso di parentesi

(**) L'operatore ternario "?:" verrà illustrato nell'Unità "Strutture di controllo".

M. Malatesta 4-Operare con i dati-02

20
09/03/2014

Modificatori dei tipi di dati

Esistono delle parole chiave che modificano il tipo di dato.

- **unsigned**
Per gli interi consente capacità doppia perché il segno non viene usato. Il default è **signed**
- **short**
Per gli interi usa una quantità di memoria pari alla metà (maggiore velocità nei calcoli)

Conversione di tipo

```
# include <stdio.h>
int main()
{
    float x=5.8;
    int y=3,
        prodotto;
    prodotto=x*y;
    printf("%f * %d = %d\n", x, y, prodotto);
    getchar();
    return 0;
}
```

La conversione implicita segue la **regola del più forte** indicata di seguito:

long double → double → float → long → int

Tra due qualunque tipi, il risultato sarà del tipo più a sinistra

In questo caso il risultato del prodotto di un **int** e di un **float** sarebbe **float**, ma essendo la variabile *prodotto* di tipo **int** il risultato (che è 17.4) viene troncato dal compilatore scartando la parte frazionaria. Questo tipo di conversione si dice **conversione implicita** (*casting* implicito) analoga a quella che avviene tra i caratteri e gli interi. Dichiarando anche *prodotto* come **float** si ottiene un risultato corretto.

Conversione di tipo

```
# include <stdio.h>
int main()
{
    float x=5.8;
    int y=3,
        prodotto;
    prodotto=(int) x*y;
    printf("%d * %d = %d\n", x, y, prodotto);
    getchar();
    return 0;
}
```

La variabile *prodotto* viene calcolata facendo precedere il valore di *x* da una conversione a **int**. Il risultato sarà un intero (15).
La conversione del tipo di *x* è fatta con l'espressione **(int) x**

Questo tipo di conversione si dice **conversione esplicita** (*casting esplicito*) ed è imposta dal programmatore. Essa si realizza con l'operatore **cast** che prevede la sintassi
(tipo) espressione
dove *tipo* è il tipo di arrivo in cui viene convertita *espressione*

La funzione rand()

```
#include <stdio.h>
#include <math.h>
#include <time.h>
int main()
{ int d1, d2;
  time_t t1, t2;
  t2=time(&t1);
  srand(t2);
  d1=rand()%6+1;
  d2=rand()%6+1;
  printf("(%d , %d) -> %d", d1, d2, d1+d2);
  getchar();
  return 0;
}
```

Il tipo di dato **time_t** viene usato per variabili destinate a contenere i valori dell'orologio di sistema.

In questo caso, il generatore di numeri casuali **srand()** viene inizializzato ad ogni esecuzione con un valore diverso (orologio di sistema) in modo da ottenere sempre numeri diversi.

Questa relazione garantisce che il valore ottenuto sia tra 1 e 6.

La funzione **sizeof()**

```
#include <stdio.h>
int main()
{
    int i;
    float f;
    char c;
    double d;
    printf("char: %d byte\n", sizeof (c));
    printf("int: %d byte\n", sizeof (i));
    printf("float: %d byte\n", sizeof (f));
    printf("double: %d byte\n", sizeof (d));
    getchar();
    return 0;
}
```

La funzione **sizeof** (*tipo*) restituisce il numero di byte del dato *tipo*.

Argomenti

- Operatori
- Assegnamento
- Operatori aritmetici interi
- Operatori aritmetici reali
- Operatori relazionali
- Operatori logici
- Tabella di verità
- Operatori sul bit
- Operatori abbreviati
- Definizione di espressione
- Esempi di espressioni
- Precedenza degli operatori
- Modificatori dei tipi di dati
- Conversione di tipo
- La funzione **rand()**
- La funzione **sizeof()**

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 4-Operare con i dati-02

27
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 4-Operare con i dati-02

28
09/03/2014