



PEGASO
Università Telematica



Indice

1. ITERAZIONI CONTROLLATE DA CONTATORE	3
BIBLIOGRAFIA	7

1. Iterazioni controllate da contatore

Considerate il seguente problema:

Una classe di dieci studenti fa un quiz.

Avete a disposizione i voti (numeri interi compresi tra 0 e 100) per questo quiz. Determinate la media della classe in riferimento al quiz.

La media della classe è uguale alla somma dei voti divisa per il numero degli studenti. L'algoritmo per risolvere questo problema su un computer deve ricevere in ingresso ognuno dei voti, eseguire il calcolo della media e stampare il risultato.

Usiamo l'iterazione controllata da contatore per inserire i voti uno alla volta.

Questa tecnica usa una variabile chiamata contatore per specificare il numero delle volte in cui un insieme di istruzioni deve essere eseguito.

In questo esempio l'iterazione termina quando il contatore supera il valore 10.

Usiamo lo pseudocodice per elencare le azioni da eseguire e specificare l'ordine in cui devono essere eseguite.

L'iterazione controllata da contatore è spesso chiamata *iterazione definita*, perché il numero delle iterazioni si conosce prima che il ciclo inizi l'esecuzione.

- 1 Poni il totale uguale a zero
- 2 Poni il contatore dei voti uguale a uno
- 3
- 4 Finché il contatore dei voti è minore o uguale a dieci
- 5 Leggi il voto successivo
- 6 Somma il voto al totale
- 7 Somma uno al contatore dei voti
- 8
- 9 Poni la media della classe uguale al totale diviso per dieci
- 10 Stampa la media della classe

```
01 // 09_03.c
02 // Media di una classe con l'iterazione controllata da contatore.
03 #include <stdio.h>
04
05 // la funzione main inizia l'esecuzione del programma
06 int main( void )
07 {
08     unsigned int counter; // numero del prossimo voto da inserire
09     int grade; // valore del voto
10     int total; // somma dei voti inseriti dall'utente
11     int average; // media dei voti
12
13     // fase di inizializzazione
14     total = 0; // inizializza il totale
15     counter = 1; // inizializza il contatore del ciclo
16
17     // fase di elaborazione
18     while ( counter <= 10 ) { // ripeti 10 volte
19         printf( "%s", "Enter grade: " ); // prompt per l'ingresso
20         scanf( "%d", &grade ); // leggi il voto
21         total = total + grade; // somma il voto al totale
22         counter = counter + 1; // incrementa il contatore
23     } // fine di while
24
25     // fase di terminazione
26     average = total / 10; // divisione intera
27
28     printf( "Class average is %d\n", average ); // stampa il risultato
```

```
29 } // fine della funzione main
```

```
Enter grade: 98
Enter grade: 76
Enter grade: 71
Enter grade: 87
Enter grade: 83
Enter grade: 90
Enter grade: 57
Enter grade: 79
Enter grade: 82
Enter grade: 94
Class average is 81
```

L'algoritmo menziona un totale e un contatore.

Un totale è una variabile usata per accumulare la somma di una serie di valori.

Un contatore è una variabile (riga 8) usata per contare (in questo caso, per contare il numero dei voti inseriti).

Poiché la variabile per il contatore è usata in questo programma per contare da 1 a 10 (tutti valori positivi), abbiamo dichiarato la variabile come tipo **unsigned int**, che può memorizzare soltanto valori non negativi (cioè 0 e oltre).

Le variabili usate per memorizzare i totali devono normalmente essere inizializzate a zero prima di essere usate in un programma, altrimenti la somma includerebbe anche il precedente valore memorizzato nella locazione di memoria del totale.

Le variabili contatore sono normalmente inizializzate a zero o a uno, a seconda del loro uso (presenteremo esempi di ognuno dei due casi).

Una variabile non inizializzata contiene un valore "spazzatura", l'ultimo valore memorizzato nella locazione di memoria riservata per quella variabile.

☹ *Errore comune di programmazione*

Se un contatore o un totale non viene inizializzato, i risultati del vostro programma saranno probabilmente scorretti. Questo è un esempio di errore logico.

😊 *Prevenzione di errori*

Inizializzare tutti i contatori e i totali.

Il calcolo della media nel programma produce come risultato il numero intero 81. In realtà, la somma dei voti in questo esempio è 817, che diviso per 10 produce il valore 81,7, cioè un numero con la virgola. Vedremo come trattare tali numeri (chiamati numeri in virgola mobile) nel prossimo paragrafo.

Nota sul collocamento delle definizioni di variabili

Il C standard consente di collocare dovunque dentro main ciascuna definizione di variabile prima del primo uso di quella variabile nel codice.

In questo documento continuiamo a raggruppare le nostre definizioni di variabili all'inizio di main per enfatizzare l'inizializzazione, l'elaborazione e le fasi di chiusura di programmi semplici. In generale, collocheremo ciascuna definizione di variabile subito prima del primo uso di quella variabile. Questa pratica aiuti a ridurre gli errori.

Bibliografia

- Paul Deitel, Harvey Deitel, "Il linguaggio C – Fondamenti e tecniche di programmazione", Libro edito da Pearson Italia. Include anche utili esercizi di autovalutazione.