
Prefazione

Benvenuti nel mondo del C. Questo libro è stato scritto da un padre, Harvey M. Deitel, e da suo figlio, Paul J. Deitel. Il padre ha programmato e/o insegnato la programmazione per 38 anni, mentre il figlio ha programmato e/o insegnato la programmazione per 18. Il padre programma e insegna grazie all'esperienza maturata; il figlio, invece, programma e insegna grazie a un'inesauribile riserva di energia. Il padre punta alla chiarezza, il figlio alle prestazioni. Il padre cerca l'eleganza e la bellezza, il figlio vuole soprattutto risultati concreti. Insieme hanno cercato di realizzare un libro che speriamo possiate trovare utile, completo e divertente.

Lo scopo di questo libro

Il Professor Harvey M. Deitel ha tenuto corsi universitari introduttivi sulla programmazione per 20 anni, puntando soprattutto a insegnare come sia possibile sviluppare programmi ben scritti e ben strutturati. Molti dei suoi insegnamenti riguardano i concetti fondamentali della programmazione, con particolare enfasi sull'utilizzo efficace delle strutture di controllo e sulla funzionalità. Questi argomenti vengono presentati all'interno di questo libro esattamente nel modo in cui il Professor Deitel li ha sempre proposti ai propri studenti universitari.

In base alla nostra esperienza, gli studenti affrontano i temi presentati in questi capitoli nello stesso modo in cui affrontano i corsi di Pascal. C'è, comunque, un'importante differenza: gli studenti sono estremamente motivati dal fatto che stanno imparando un linguaggio che potranno utilizzare immediatamente quando lasceranno l'università e affronteranno il mondo del lavoro. Tutto questo aumenta notevolmente il loro entusiasmo nei confronti del C, nonostante ci sia moltissimo da imparare.

Il nostro obiettivo era chiaro: offrire un testo sul C destinato ai corsi universitari a livello introduttivo, per quegli studenti che non hanno alcuna esperienza nell'ambito della programmazione, pur fornendo, allo stesso tempo, la completezza teorica e pratica richiesta ai tradizionali corsi avanzati di C.

Il testo segue lo standard ANSI C; tenete presente che molte funzionalità dell'ANSI C non sono implementate nelle versioni pre-ANSI del C. Per avere informazioni più dettagliate sul linguaggio vi conviene consultare il manuale di riferimento del vostro sistema o procurarvi una copia del documento ANSI/ISO 9899: 1990, "American National Standard for Information Systems-Programming Language C", dell'American National Standards Institute, 11 West 42nd Street, New York, New York 10036.

La metodologia di insegnamento

Questo libro contiene un'ampia varietà di esempi, esercizi e progetti che prendono spunto da molte situazioni e che offrono agli studenti la possibilità di risolvere problemi reali. Il libro prende in esame i principi della progettazione del software, insistendo sull'importanza della chiarezza dei programmi ed evitando l'uso di terminologia complessa a favore di esempi chiari e diretti, il cui codice sia stato collaudato sulle piattaforme C più diffuse.

L'apprendimento attraverso il codice

Il libro presenta una grande quantità di esempi basati sul codice. Ogni argomento viene presentato nell'ambito di un programma C completo e funzionante, seguito sempre da una o più finestre che mostrano l'output del programma in questione. Viene quindi usato il linguaggio per insegnare il linguaggio, e la lettura di questi programmi offre un'esperienza molto simile alla loro esecuzione reale su di un computer.

L'accesso al World Wide Web

Tutto il codice presente nel libro si trova anche in Internet, nel booksite abbinato a questo libro, all'indirizzo <http://www.apogeeonline.com.education.booksite>. Il nostro consiglio è quello di scaricare tutto il codice, eseguendo poi ogni singolo programma via via che appare nel testo. Potete anche modificare il codice degli esempi e vedere cosa succede, imparando così a programmare programmando. Tutto questo materiale è protetto da diritti d'autore, quindi utilizzatelo liberamente per studiare il C, ma non pubblicatene alcuna parte senza l'esplicito permesso da parte degli autori e della casa editrice.

Obiettivi

Ogni capitolo inizia con la presentazione degli *Obiettivi*. Gli studenti possono così sapere in anticipo ciò che andranno ad apprendere e, alla fine della lettura del capitolo, potranno verificare se hanno raggiunto o meno questi obiettivi.

Il codice e gli esempi

Le funzionalità del C vengono presentate nell'ambito di programmi completi e funzionanti. Ogni programma è seguito dalle immagini degli output che vengono prodotti, così che gli studenti possano assicurarsi della correttezza dei risultati. I programmi presentati vanno da poche linee di codice a esempi composti da varie centinaia di righe. Gli studenti dovrebbero scaricare tutto il codice dal sito Web <http://www.apogeeonline.com/education/booksite>, eseguendo poi ogni programma via via che questo viene presentato all'interno del testo.

Figure e immagini

Il libro offre un'ampia varietà di grafici, immagini e output di programmi. Nelle sezioni dedicate alle strutture di controllo, per esempio, appaiono vari diagrammi di flusso molto utili. [Nota: I diagrammi di flusso non vengono presentati come uno strumento di sviluppo, ma ricorriamo a una breve presentazione basata su questi diagrammi per specificare le singole operazioni di ogni struttura di controllo C.]

Consigli e suggerimenti

Il libro offre molti suggerimenti riguardanti la programmazione, per aiutare gli studenti a concentrarsi sugli aspetti più importanti dello sviluppo di programmi. Questi consigli vengono forniti attraverso le sezioni chiamate *Buona abitudine*, *Errore tipico*, *Collaudo e messa a punto*, *Obiettivo efficienza*, *Obiettivo portabilità*, *Ingegneria del software*.



Buona abitudine

Il concetto più importante per chi inizia a programmare è la chiarezza, e all'interno delle sezioni *Buona abitudine* vengono presentate delle tecniche per la scrittura di programmi chiari, comprensibili e più facilmente gestibili.



Errore tipico

Tutti gli studenti che affrontano per la prima volta un nuovo linguaggio tendono a commettere frequentemente gli stessi errori. Le sezioni *Errore tipico* aiutano gli studenti a evitare di commettere i più comuni.



Collaudo e messa a punto

Queste sezioni forniscono consigli circa le attività di test e debugging dei programmi C, anche a livello preventivo.



Obiettivo efficienza

In base alla nostra esperienza, insegnare agli studenti come scrivere programmi chiari e comprensibili deve costituire l'obiettivo più importante di qualsiasi corso di programmazione. Gli studenti, però, vogliono normalmente imparare a scrivere programmi che vengano eseguiti in modo veloce, che utilizzino poca memoria, che richiedano una minima quantità di comandi e che offrano prestazioni eccellenti. Le sezioni *Obiettivo efficienza* offrono suggerimenti su come migliorare le prestazioni dei propri programmi.



Obiettivo portabilità

Alcuni programmatori pensano che, implementando un'applicazione C, questa sia immediatamente portabile su tutte le piattaforme; sfortunatamente, non è sempre così. Le sezioni *Obiettivo portabilità* aiutano gli studenti a scrivere codice realmente portabile, fornendo inoltre informazioni su come il C sia in grado di raggiungere questo elevato livello di portabilità.



Ingegneria del software

Il C è un linguaggio molto efficace nell'ambito della progettazione del software, e le sezioni *Ingegneria del software* prendono in esame gli aspetti architettonici e progettuali che influiscono sulla realizzazione di sistemi software, specialmente nel caso di sistemi su vasta scala. Molte di queste informazioni saranno utili agli studenti nei corsi più avanzati, oltre che nel mondo del lavoro.

Esercizi di autovalutazione

Il libro propone molti esercizi corredati di risposte, così che gli studenti possano prepararsi alle esercitazioni vere e proprie. Gli studenti dovrebbero essere incoraggiati a svolgere tutti questi esercizi di autovalutazione.

Esercizi

Ogni capitolo si conclude con un insieme di esercizi di vario tipo, che permettono agli insegnanti di adattare le proprie lezioni alle esigenze particolari di ogni classe. Gli esercizi verificano l'apprendimento dei concetti e dei termini più importanti, chiedono agli studenti di scrivere singole istruzioni, piccole porzioni di funzioni, funzioni e programmi completi fino ad arrivare alla costruzione di interi progetti.

Indice analitico

Alla fine del libro è possibile trovare un indice analitico completo, molto utile sia a chi legge questo libro per la prima volta che ai programmatori che lo usano come riferimento.

Panoramica sul libro

Capitolo 1 - Nozioni sulla elaborazione elettronica: spiega cosa sono i computer, come funzionano e in che modo possono essere programmati. Questo capitolo introduce la programmazione strutturata e spiega perché questo insieme di tecniche abbia favorito una rivoluzione nel modo di scrivere i programmi. Il capitolo offre una breve storia dello sviluppo dei linguaggi di programmazione, dai linguaggi macchina, ai linguaggi assembly, ai linguaggi di alto livello. È discussa anche l'origine del linguaggio di programmazione C. Il capitolo include un'introduzione all'ambiente di programmazione del C.

Capitolo 2 - Introduzione alla programmazione in C: fornisce un'introduzione sintetica ai programmi di scrittura C e offre un trattamento dettagliato delle operazioni decisionali e aritmetiche del C. Dopo aver studiato questo capitolo, lo studente avrà imparato a scrivere programmi semplici ma completi.

Capitolo 3 - Lo sviluppo di programmi strutturati: è probabilmente il capitolo più importante del testo, specialmente per un diligente studente di informatica. Esso introduce la nozione di algoritmi (procedure) per la risoluzione dei problemi; spiega l'importanza della programmazione strutturata per la produzione di programmi comprensibili, che siano collaudabili e modificati e che abbiano più probabilità di funzionare correttamente sin dal primo tentativo. Introduce anche le strutture di controllo fondamentali della programmazione strutturata, cioè le strutture di sequenza, selezione (**if** e **if/else**) e di iterazione (**while**). Il capitolo spiega la programmazione top-down a raffinamenti successivi, una tecnica importante per produrre in modo corretto programmi strutturati. Esso presenta un popolare strumento di aiuto alla progettazione dei programmi, ovvero sia lo pseudocodice strutturato. I metodi e gli approcci usati nel Capitolo 3 sono applicabili alla programmazione strutturata in ogni linguaggio di programmazione, non solo a quella in C. Questo capitolo aiuta lo studente a sviluppare delle buone abitudini di programmazione che lo prepareranno ad affrontare gli esercizi di programmazione più consistenti del resto del libro.

Capitolo 4 - Il controllo del programma: raffina le nozioni di programmazione strutturata e introduce altre strutture di controllo; esamina in dettaglio l'iterazione e paragona i cicli controllati da un contatore con quelli controllati da un valore sentinella. La struttura **for** è presentata come uno strumento adatto a implementare i cicli controllati da un contatore. Sono introdotte anche la struttura di selezione **switch** e quella di iterazione **do/while**. Il capitolo termina con una discussione sugli operatori logici.

Capitolo 5 - Le funzioni: tratta della progettazione e costruzione dei moduli del programma. Il linguaggio C include funzioni di libreria standard, funzioni definite dal programmatore, la ricorsione e la possibilità di effettuare delle chiamate per valore. Le tecniche presentate sono essenziali per la produzione e la comprensione di programmi strutturati, specialmente di quelli più grandi, e del software che i programmatori di sistemi e di applicazioni svilupperanno più probabilmente nelle applicazioni del mondo reale. La strategia "dividi e conquista" è presentata come un mezzo efficace per risolvere i problemi più complessi; le funzioni aiutano il programmatore a suddividere i programmi complessi in componenti più semplici che interagiscono tra di loro. Agli studenti piace avere a che fare con i numeri casuali e le simulazioni e apprezzano la trattazione del gioco di dadi craps che fa un uso elegante delle strutture di controllo. Il capitolo offre una solida introduzione alla ricorsione e include una tabella che riassume i 31 esempi ed esercizi sulla ricorsione distribuiti nel resto del libro. Alcuni libri affrontano la ricorsione solo verso la loro fine, ma noi pensiamo che sia meglio trattare l'argomento in modo graduale nel corso di tutto il libro. L'esteso gruppo di 39 esercizi presentato alla fine del Capitolo include vari problemi classici sulla ricorsione come le Torre di Hanoi.

Capitolo 6 - I vettori: discute la strutturazione dei dati in vettori, cioè gruppi di dati correlati e dello stesso tipo. Il capitolo presenta numerosi esempi sia sui vettori con un solo indice che su quelli con indice doppio. L'importanza della strutturazione dei dati è ampiamente riconosciuta al pari dell'utilizzo delle strutture di controllo nello sviluppo di programmi strutturati. Alcuni esempi del capitolo approfondiscono varie tipiche elaborazioni dei vettori, la stampa degli istogrammi, la classificazione dei dati, il passaggio dei vettori alle funzioni e un'introduzione all'analisi dei sondaggi di opinione. Un punto caratteristico di questo capitolo è un'accurata presentazione della ricerca binaria, un enorme miglioramento di quella lineare. Gli esercizi alla fine del capitolo includono una vasta selezione di problemi interessanti e stimolanti. Questi includono tecniche di ordinamento migliorate, la progettazione di un sistema di prenotazioni per linee aeree, un'introduzione al concetto della Turtle Graphic (resa famosa dal linguaggio LOGO) e i problemi del giro del cavallo e delle otto regine che introducono la nozione di programmazione euristica, molto utilizzata nel campo dell'intelligenza artificiale.

Capitolo 7 - I puntatori: presenta una delle caratteristiche più potenti del linguaggio C. Il capitolo fornisce spiegazioni dettagliate sugli operatori dei puntatori, sulla chiamata per riferimento, sulle espressioni con puntatori, sull'aritmetica dei puntatori, sulla relazione tra puntatori e vettori, sui vettori di puntatori e sui puntatori a funzioni. Gli esercizi del capitolo includono una simulazione della classica gara tra la tartaruga e la lepre e vari algoritmi per il mescolamento e la distribuzione delle carte. È inclusa anche una speciale sezione intitolata "Costruite il vostro computer", che spiega la programmazione in linguaggio macchina e prosegue con un progetto che comprende la progettazione e l'implementazione di un simulatore che permetta al lettore di scrivere e far funzionare i programmi sviluppati in linguaggio macchina. Questa caratteristica unica del libro sarà utile soprattutto

to al lettore che vorrà capire il reale funzionamento dei computer. Ai nostri studenti piace questo progetto e spesso implementano sostanziali miglioramenti; molti di questi sono proposti all'interno degli esercizi. Nel Capitolo 12, un'altra sezione speciale guiderà il lettore attraverso la costruzione di un compilatore; il linguaggio macchina prodotto dal compilatore sarà poi eseguito con il simulatore di linguaggio macchina prodotto nel Capitolo 7.

Capitolo 8 - I caratteri e le stringhe: tratta dei principi fondamentali dell'elaborazione dei dati non numerici. Il capitolo include una panoramica completa delle funzioni per l'elaborazione dei caratteri e delle stringhe disponibili nella libreria del C. Le tecniche discusse in questo contesto sono ampiamente utilizzate nello sviluppo di word processor, software di impaginazione e applicazioni per l'elaborazione del testo. Allo studente piaceranno gli esercizi sui limerick, sulla generazione casuale di poesie, sulla conversione dall'inglese al pig Latino, sulla generazione di parole di sette lettere che equivalgono a un numero telefonico dato, sull'allineamento del testo, sulla protezione degli disegni, sulla scrittura della cifra di un disegno in lettere, sulla generazione del Codice Morse, sulle conversioni metriche e sulle lettere di sollecito. L'ultimo esercizio invita lo studente ad utilizzare un dizionario computerizzato per creare un generatore di cruciverba!

Capitolo 9 - La formattazione dell'Input/Output: presenta tutte le potenti capacità di formattazione di printf e scanf. Discuteremo delle capacità di formattazione dell'output di printf, come l'arrotondamento dei valori in virgola mobile a un dato numero di cifre decimali, l'incolonnamento dei numeri, l'allineamento a destra e a sinistra, l'inserimento di informazioni letterali, la forzatura del segno positivo, la stampa degli zeri iniziali, l'uso della numerazione esponenziale, l'uso dei numeri ottali ed esadecimali e il controllo delle dimensioni di campo e della precisione. Discuteremo tutte le sequenze di escape di printf per il movimento del cursore, per la stampa dei caratteri speciali e per la produzione di un allarme acustico. Esamineremo tutte le possibilità di formattazione dell'input di scanf incluso l'input di specifici tipi di dati e come ignorare dei caratteri specifici presenti sullo stream di input. Discuteremo tutte le specifiche di conversione di scanf per leggere i numeri decimali, ottali, esadecimali, in virgola mobile, i caratteri ed i valori di una stringa. Discuteremo della scansione dell'input in modo che corrisponda (o meno) ai caratteri inclusi in un gruppo di scansione. Gli esercizi del capitolo verificano virtualmente tutte le capacità di input/output formattato.

Capitolo 10 - Le strutture, le unioni, le elaborazione dei bit e l'enumerazione: presenta diverse caratteristiche importanti. Le strutture sono come i record del Pascal e di altri linguaggi: esse raggruppano dati di tipi diversi. Le strutture sono usate nel Capitolo 11 per formare file composti da record di informazioni. Nel Capitolo 12 le strutture sono usate insieme ai puntatori e all'allocazione dinamica della memoria, per formare strutture dinamiche di dati come le liste concatenate, le code, le pile e gli alberi. Le unioni consentono di utilizzare un'area di memoria per diversi tipi di dato in tempi differenti; questa condivisione può ridurre la quantità di memoria principale o secondaria richiesta da un programma. Le enumerazioni forniscono un modo conveniente per definire delle costanti simboliche utili; ciò aiuta a rendere i programmi più autoesplicativi. Le potenti capacità di manipolazione del bit del C consentono ai programmatori di scrivere programmi che usano le capacità a basso livello dell'hardware. Ciò aiuta i programmi ad elaborare stringhe di bit, ad impostare a on o a off i singoli bit e ad immagazzinare le informazioni in modo più compatto. Tali capacità, spesso disponibili solo nei linguaggi assembly di basso livello,

sono particolarmente apprezzate dai programmatori nella scrittura di software per i sistemi operativi e per la gestione delle reti. Una caratteristica di questo capitolo è la nuova versione ad alta efficienza della simulazione del mescolatore e distributore di carte. Questa è una eccellente opportunità per l'insegnante per porre l'accento sulla qualità degli algoritmi.

Capitolo 11 - L'elaborazione dei file: discute delle tecniche usate per elaborare i file di testo ad accesso sequenziale e casuale. Il capitolo inizia con un'introduzione alla gerarchia dei dati: dai bit ai byte, ai campi, ai record, ai file. Successivamente è presentata una semplice panoramica sui file e sugli stream del C. I file ad accesso sequenziale sono discussi usando una serie di tre programmi che mostrano come aprire e chiudere i file, come immagazzinare i dati in modo sequenziale in un file e come leggere in modo sequenziale i dati da un file. Si discute dei file ad accesso casuale usando una serie di quattro programmi che mostrano come creare un file ad accesso casuale, come leggere e scrivere i dati con accesso casuale, e come leggere i dati in modo sequenziale da un file ad accesso casuale. Il quarto programma combina molte tecniche di accesso ai file, sia sequenziali che casuali, in un programma completo per l'elaborazione di transazioni. Gli studenti dei nostri seminari industriali ci hanno detto che dopo aver studiato il materiale sull'elaborazione dei file, sono stati capaci di produrre programmi corposi per l'elaborazione dati che sono stati immediatamente utili all'interno delle loro aziende.

Capitolo 12 - Le strutture di dati: discute le tecniche usate per creare delle strutture dinamiche di dati. Il capitolo inizia discutendo delle strutture ricorsive e dell'allocatione dinamica della memoria e prosegue con una discussione su come creare e conservare varie strutture dinamiche di dati come le liste concatenate, le code (o liste di attesa), le pile e gli alberi. Per ogni tipo di struttura di dati presentiamo dei programmi funzionanti e completi e mostriamo degli esempi di output. Il Capitolo 12 consente allo studente di padroneggiare effettivamente i puntatori. Il capitolo include molti esempi di utilizzo dell'indirizione e della doppia indirizione (un concetto particolarmente difficile). Uno dei problemi legati all'utilizzo dei puntatori è che gli studenti hanno difficoltà a visualizzare le strutture di dati e il modo in cui i loro nodi sono collegati, così abbiamo incluso delle illustrazioni che mostrano i collegamenti e la sequenza in cui essi sono creati. L'esempio dell'albero binario è una stupenda pietra miliare per lo studio dei puntatori e delle strutture dinamiche di dati. Questo esempio crea un albero binario, si preoccupa dell'eliminazione dei duplicati e introduce l'attraversamento ricorsivo dell'albero in ordine anticipato, in ordine e in ordine posticipato. Gli studenti ritengono realmente esaustivo lo studio e l'implementazione di questo esempio e apprezzano in modo particolare il vedere che l'attraversamento in ordine dell'albero stampa i valori dei nodi nell'ordine stabilito. Il capitolo include un gruppo consistente di esercizi. Rilevante è la sezione speciale "Costruite il vostro compilatore". Gli esercizi accompagnano lo studente nello sviluppo di un programma di conversione dalla notazione infissa a quella polacca inversa e di un programma di valutazione di espressioni in notazione polacca inversa. Abbiamo quindi modificato l'algoritmo di valutazione di espressioni in notazione polacca inversa in modo da poter generare un codice in linguaggio macchina. Il compilatore immagazzina il suddetto codice in un file (usando le tecniche del capitolo 11). Gli studenti eseguono quindi il codice in linguaggio macchina prodotto dai loro compilatori sul simulatore di software costruito negli esercizi del Capitolo 7!

Capitolo 13 - Il preprocessore: offre una discussione dettagliata delle direttive del preprocessore. Il capitolo include informazioni più complete sulla direttiva `#include` che

induce l'inclusione di una copia di un file specificato in sostituzione della direttiva, prima che il file sia compilato, e sulla direttiva **#define** che definisce delle costanti simboliche e delle macro. Il capitolo spiega la compilazione condizionata per consentire al programmatore di controllare l'esecuzione delle direttive del preprocessore e la compilazione del codice del programma. Si parla dell'operatore **#** che converte il suo operando in una stringa e dell'operatore **##** che concatena due token. Sono presentate le cinque costanti simboliche (**__LINE__**, **__FILE__**, **__DATE__**, **__TIME__** e **__STDC__**). Alla fine è trattata la macro **assert** del file di intestazione **assert.h**. Assert è preziosa nel collaudo, la messa a punto, la verifica e la convalida del programma.

Capitolo 14 - Argomenti avanzati: presenta vari argomenti avanzati che di solito non sono affrontati nei corsi introduttivi. La sezione 14.2 mostra come redirezionare l'input di un programma in modo che provenga da un file, come redirezionare l'output in modo che sia immagazzinato in un file, come redirezionare l'output di un programma in modo che vada a costituire l'input di un altro programma (piping) e come accodare l'output di un programma ad un file già esistente. La sezione 14.3 discute di come sviluppare funzioni che usano elenchi variabili di argomenti. La sezione 14.4 mostra in che modo gli argomenti della riga di comando possano essere passati alla funzione main e come utilizzati all'interno di un programma. La sezione 14.5 tratta della compilazione di programmi i cui componenti possono essere distribuiti in vari file. La sezione 14.6 discute della registrazione delle funzioni con atexit in modo che possano essere eseguite al termine dell'esecuzione di un programma e come terminare l'esecuzione di un programma con la funzione exit. La sezione 14.7 discute dei qualificatori di tipo const e volatile. La sezione 14.8 mostra come specificare il tipo di una costante numerica, usando dei suffissi per gli interi e i numeri in virgola mobile. La sezione 14.9 tratta dei file binari e dell'uso dei file temporanei. La sezione 14.10 mostra come usare la libreria per la gestione dei segnali per intercettare degli eventi inattesi. La sezione 14.11 discute della creazione e dell'uso dei vettori dinamici con le funzioni calloc e realloc.

Molte Appendici forniscono un valido materiale di riferimento. In particolare, nell'Appendice A forniamo la sintassi del linguaggio C; nell'Appendice B un elenco commentato di tutte le funzioni della libreria standard del C; nell'Appendice C una tabella completa della priorità e dell'associatività degli operatori; nell'Appendice D l'insieme dei codici dei caratteri ASCII e, nell'Appendice E, una discussione sui sistemi numerici binari, ottali, decimali ed esadecimali. L'Appendice B è stata estratta dal documento dell'ANSI standard con l'esplicito permesso scritto dell'American National Standards Institute; la suddetta appendice fornisce un riferimento valido e dettagliato per l'esercitazione del programmatore C. L'Appendice E è una spiegazione completa dei sistemi numerici, che include molti esercizi di auto-valutazione e le relative risposte.

Vi preghiamo di inviarci i vostri commenti, suggerimenti, critiche e correzioni, al fine di migliorare ulteriormente questo libro. L'indirizzo a cui scriverci è il seguente:

deitel@deitel.com

Bene, per ora è tutto. Vi rinnoviamo il benvenuto nel mondo del C e dello sviluppo delle applicazioni del futuro. Buona fortuna e buon lavoro!

Harvey M. Deitel - Paul J. Deitel