

- i) Nei programmi in C si punta sempre ai file con il loro nome. *F CON UN PUNTERE*  
 j) Ogni volta che un programma crea un file, questo sarà conservato automaticamente dal computer per i riferimenti futuri. ✓

11.7 L'Esercizio 11.3 ha chiesto al lettore di scrivere una serie di istruzioni distinte. In realtà, quelle istruzioni formano il nucleo di un importante tipo di programma per l'elaborazione dei file, ovvero sia, un programma di file-matching (confronto di file). Nelle elaborazioni dei dati commerciali, è uso comune avere diversi file in ogni sistema. In un sistema di gestione dei crediti, per esempio, c'è generalmente un file principale che contiene informazioni dettagliate su ogni cliente, come il suo nome, l'indirizzo, il numero di telefono, gli insoluti, il limite di credito, le condizioni di sconto, le disposizioni contrattuali e possibilmente un succinto riepilogo dei suoi recenti acquisti e pagamenti.

Man mano che le transazioni verranno concluse (ovverosia, verranno effettuate delle vendite e arriveranno dei pagamenti nella corrispondenza), queste saranno immesse in un file. Alla fine di ogni periodo commerciale (un mese per alcune aziende, una settimana per altre, un giorno in certi casi) il file delle transazioni (chiamato **"trans.dat"** nell'Esercizio 11.3) sarà confrontato con il file principale (chiamato **"oldmast.dat"** nell'Esercizio 11.3), aggiornando così tutti i record di conto con gli ultimi acquisti e pagamenti. Dopo che i suddetti aggiornamenti saranno stati eseguiti, il file principale sarà riscritto in un nuovo archivio (**"newmast.dat"**) che, alla fine del successivo periodo commerciale, sarà usato per ricominciare il processo di aggiornamento.

I programmi di file-matching devono affrontare alcuni problemi che non hanno riscontro in quelli che gestiscono degli archivi singoli. Per esempio, non sempre sarà possibile riscontrare delle corrispondenze tra i record dei due file. Un cliente presente sul file principale potrebbe non aver effettuato alcun acquisto o pagamento nel periodo commerciale corrente e, di conseguenza, per questo cliente non esisterebbe nessun record nel file delle transazioni. Allo stesso modo, un cliente che abbia effettuato qualche acquisto o pagamento potrebbe essere appena giunto nella comunità, e l'azienda potrebbe non aver avuto l'opportunità di creare per lui un record principale.

Usate le istruzioni scritte nell'Esercizio 11.3 come base per scrivere un programma di file-matching per la contabilità del credito. In ognuno dei file, usate il numero di conto come chiave del record per riscontrare le corrispondenze. Supponete che siano tutti file sequenziali con record immagazzinati in ordine crescente di numero di conto.

Ogni volta che avrete riscontrato una corrispondenza (ovverosia, quando avrete trovato due record con lo stesso numero di conto sul file principale e su quello delle transazioni), dovrete aggiungere al saldo corrente del file principale l'ammontare in dollari rilevato dall'archivio delle transazioni e dovrete scrivere il record di **"newmast.dat"**. Considerate che gli acquisti saranno indicati sul file delle transazioni con importi positivi, mentre i pagamenti saranno indicati con un ammontare negativo. Nel caso in cui per un certo conto troviate un record principale, ma nessuna corrispondenza nelle transazioni, dovrete semplicemente scrivere il record principale in **"newmast.dat"**. Nel caso in cui troviate un record di transazione, ma nessuna corrispondenza tra quelli principali, dovrete visualizzare il messaggio **"Unmatched transaction record for account number ..."** (completate il messaggio, rilevando il numero di conto dal record di transazione).

11.8 Dopo aver scritto il programma dell'Esercizio 11.7, scrivetene uno semplice che crei alcuni dati di prova per controllare il programma dell'Esercizio 11.7. Usate i seguenti dati di conto:

File principale:		
Numero di conto	Nome	Saldo
100	Alan Jones	348.17
300	Mary Smith	27.19
500	Sam Sharp	0.00
700	Suzy Green	-14.22



**File delle transazioni:**

Numero di conto	Importo in dollari
100	27.14
300	62.11
400	100.56
900	82.17

**11.9** Eseguite il programma dell'Esercizio 11.7 usando i file di prova creati nell'Esercizio 11.8. Usate il programma della Sezione 11.7 per visualizzare un nuovo file principale. Verificate attentamente i risultati.

**11.10** È probabile (anzi molto comune) che ci siano diversi record di transazione con la stessa chiave. Ciò potrebbe accadere perché, durante un periodo commerciale, un particolare cliente potrebbe effettuare diversi acquisti e pagamenti. Riscrivete il vostro programma di file-matching per la contabilità del credito scritto nell'Esercizio 11.7, in modo che sia possibile gestire svariati record di transazione con la stessa chiave. Modificate i dati di prova dell'Esercizio 11.8, in modo da includere questi ulteriori record di transazione:

Numero di conto	Importo in dollari
300	83.89
700	80.78
700	1.53

**11.11** Scrivete delle istruzioni che eseguano ognuno dei seguenti compiti. Supponete che la struttura

```
struct person {
    char lastName[15];
    char firstName[15];
    char age[2];
};
```

sia già stata definita e che il file sia già stato aperto in scrittura.

- Inizializzate il file "nameage.dat" in modo che ci siano 100 record con `lastName = "unassigned"`, `firstName = ""` e `age = "0"`.
- Immettete 10 cognomi, nomi ed età e scriveteli nel file.
- Aggiornate un record e; qualora non vi troviate delle informazioni, informate l'utente con il messaggio "No info".
- Eliminate un record che contenga delle informazioni, inizializzando nuovamente quel particolare record.

**11.12** Siete il proprietario di una ferramenta e avete bisogno di mantenere un inventario che possa dirvi quali e quanti attrezzi avete, e il costo di ognuno di essi. Scrivete un programma che inizializzi il file "hardware.dat" con 100 record vuoti, vi consenta di immettere i dati relativi a ogni attrezzo, vi dia la possibilità di elencarli tutti, vi lasci eliminare il record di un attrezzo che non avete più e vi permetta di aggiornare qualsiasi informazione all'interno del file. Il numero di identificazione dell'attrezzo dovrà essere anche quello del record. Usate le seguenti informazioni come dati iniziali per il vostro file.



Record n°	Nome dell'attrezzo	Quantità	Costo
3	Smerigliatrice elettrica	7	57.98
17	Martello	76	11.99
24	Sega da traforo	21	11.00
39	Falciatrice	3	79.50
56	Sega elettrica	18	99.99
68	Giravite	106	6.99
77	Martello da fabbro	11	21.50
83	Chiave inglese	34	7.50

11.13 *Generatore di parole con i numeri telefonici.* Le tastiere telefoniche standard dei cellulari contengono le cifre da zero a nove. A ognuno dei numeri da due a nove sono associate tre o quattro lettere, come indicato dalla seguente tabella:

Numero	Lettere
2	A B C
3	D E F
4	G H I
5	J K L
6	M N O
7	P Q R S
8	T U V
9	W X Y Z

Per molte persone la memorizzazione dei numeri telefonici è un'operazione difficile e, per questo motivo, utilizzano la corrispondenza tra le cifre e le lettere per sviluppare parole di sette caratteri che corrispondano ai propri numeri telefonici. Per esempio, una persona il cui numero telefonico sia 468-6374, potrebbe usare le corrispondenze indicate nella tabella precedente per sviluppare la parola di sette lettere «INUMERI».

Le aziende tentano spesso di ottenere dei numeri telefonici che possano essere facilmente ricordati dai loro clienti. Se per le telefonate dei propri clienti un'azienda potesse pubblicizzare una semplice parola, invece di un numero telefonico, allora quell'azienda riceverebbe indubbiamente qualche telefonata in più.

Ogni parola di sette lettere corrisponde esattamente a un numero telefonico di altrettante cifre. Un ristorante che desiderasse incrementare gli introiti dei propri affari, potrebbe sicuramente farlo con il numero 872-4266 (ovverosia «USCIAMO»).

A ogni numero telefonico di sette cifre corrispondono molte distinte parole di altrettante lettere. Sfortunatamente, molte di queste rappresentano giustapposizioni di lettere senza significato. Tuttavia, è probabile che il proprietario di una bottega di barbiere possa essere contento di sapere che il numero telefonico del proprio salone, 227-3554, corrisponda a «CAPELLI». Il proprietario di un negozio di alcolici sarebbe senza dubbio felice di scoprire che il numero telefonico del proprio negozio, 547-8674, coincida con «LIQUORI». Un vegetariano con il numero telefonico 837-3872 sarebbe compiaciuto di sapere che quelle cifre equivalgano alla parola «VERDURA».

Scrivete un programma C che, dato un numero di sette cifre, scriva in un file ogni possibile parola di sette lettere corrispondente a quel numero. Esistono al massimo 16384 parole possibili ( $4^7$  elevato alla settima potenza). Evitate i numeri di telefono con le cifre zero e uno.



**11.14** Nel caso in cui abbiate a disposizione un dizionario computerizzato, modificate il programma che avete scritto nell'Esercizio 11.13 in modo che ricerchi le parole nel dizionario. Alcune delle combinazioni di sette lettere create da questo programma consistono di due o più parole (il numero telefonico 646-2276 corrisponde a "MIOCAPO").

**11.15** Modificate l'esempio della Figura 8.14, in modo che utilizzi le funzioni **fgetc** e **fputs**, invece di **getchar** e **puts**. Il programma dovrà offrire all'utente la possibilità di scegliere se leggere dallo standard input e scrivere sullo standard output, o se leggere e scrivere in file specificati. Nel caso in cui l'utente scelga la seconda possibilità, consentitegli di immettere i nomi dei file di input e di output.

**11.16** Scrivete un programma che usi l'operatore **sizeof** per determinare le dimensioni in byte dei vari tipi di dato sul vostro sistema. Scrivete i risultati nel file **"datasize.dat"**, così che possiate visualizzarli in un secondo momento. Il formato dei risultati nel file dovrà essere:

Data type	Size
char	1
unsigned char	1
short int	2
unsigned short int	2
int	4
unsigned int	4
long int	4
unsigned long int	4
float	4
double	8
long double	16

Nota: sul vostro computer, le dimensioni dei tipi di dato potrebbero non corrispondere a quelle elencate qui sopra.

**11.17** Nell'Esercizio 7.19, avete scritto una simulazione software di un computer che usa uno speciale linguaggio macchina, chiamato Linguaggio Macchina Simpletron (LMS). Ogni volta che volete eseguire un programma LMS, dovete digitare da tastiera il codice per immetterlo nel simulatore. Se vi capitasse di commettere un errore durante la digitazione del programma LMS, dovrete far ripartire il simulatore e digitare nuovamente il codice LMS. Non sarebbe bello poter leggere il programma LMS da un file, piuttosto che digitarlo ogni volta? Ciò ridurrebbe il tempo e gli errori durante la fase di preparazione per l'esecuzione di un programma LMS.

- Modificate il simulatore che avete scritto nell'Esercizio 7.19, in modo che legga i programmi LMS da un file specificato tramite tastiera dall'utente.
- Al termine della sua esecuzione, il Simpletron avrà visualizzato sullo schermo il contenuto dei propri registri e della memoria. Sarebbe sicuramente bello poter catturare l'output in un file, perciò modificate il simulatore in modo che scriva il suo output in un file, oltre che sullo schermo.