

# Capitolo 2

## Biblioteca

### 2.1 Problema

*I mean your borrowers of books-those mutilators of collections, spoilers of the symmetry of shelves, and creators of odd volumes.*

*-Charles Lamb, Essays of Elia (1823) 'The Two Races of Men'*

*Parlo di quei prenditori in prestito di libri-quei mutilatori di collezioni, spogliatori della simmetria degli scaffali e creatori di volumi scompagnati. -Charles Lamb, Saggi di Elia (1823) 'Le Due Razze di Uomini'*

Come Charles Lamb, i bibliotecari hanno qualche difficoltà con la gente che prende a prestito i libri: non li rimette dove dovrebbe. Per mantenere l'ordine, i libri restituiti vengono conservati al banco del prestito fino a che un bibliotecario non ha il tempo di rimetterli al posto giusto sugli scaffali. Anche per i bibliotecari, rimettere il libro giusto al posto giusto può essere una faccenda lunga. Ma siccome molte biblioteche oggi sono informatizzate, potete scrivere un programma per aiutarli.

Quando un utente prende o restituisce un libro, il computer ne conserva traccia. I libri restituiti vengono impilati sul banco: l'ultimo rimane così in cima. Periodicamente, i bibliotecari richiedono al programma la lista dei libri restituiti e l'indicazione di dove riporre ciascun libro, così da poterli rimettere al posto giusto sugli scaffali. Il programma deve restituire la lista dei libri ancora sugli scaffali e le istruzioni che i bibliotecari devono eseguire per riporli.

#### 2.1.1 Ingresso

L'ingresso si compone di due file. Il primo elenca la dotazione della biblioteca, un libro per riga nel formato:

*autore, "titolo"*

I libri sono elencati in ordine alfabetico crescente rispetto all'autore e (in caso di autore identico oppure omonimo) rispetto al titolo. I libri sono disposti nello stesso ordine sullo scaffale della biblioteca (per semplicità, si suppone che vi sia uno scaffale solo). Si può assumere che la lunghezza di ciascuna riga (titolo e autore) non superi gli 80 caratteri e che tutti i libri abbiano autore o titolo diverso. Inizialmente, i libri sono tutti sullo scaffale. Il termine dell'inventario è indicato da una riga contenente solo la parola:

FINE

Ad esempio:

Adams S., "Il principio di Dilbert"  
Bertossi A., "Algoritmi e Strutture Dati"  
Chaucer G., "I Racconti di Canterbury"  
Hoeg P., "Il senso di Smilla per la neve"  
Hofstadter D., "Godel, Escher, Bach"  
Kernighan B. e Ritchie D., "Il linguaggio C"  
King S., "L'ombra dello scorpione"  
Pennac D., "La prosivendola"  
Shakespeare W., "Le allegre comari di Windsor"  
Yourcenar M., "L'opera al nero"  
FINE

Il secondo file contiene le registrazioni dei prestiti e delle restituzioni. Ognuna appare su una singola riga, e rispetta il formato seguente:

PRESTITO *autore, "titolo"*

RESTITUZIONE *autore, "titolo"*

La lista termina con una riga che contiene solo la parola:

FINE

Questa va interpretata dal programma come una richiesta di assistenza nel riporre i libri restituiti. Ad esempio:

PRESTITO Bertossi A., "Algoritmi e Strutture Dati"  
PRESTITO Kernighan B. e Ritchie D., "Il linguaggio C"  
PRESTITO Adams S., "Il principio di Dilbert"  
RESTITUZIONE Kernighan B. e Ritchie D., "Il linguaggio C"  
PRESTITO Shakespeare W., "Le allegre comari di Windsor"  
PRESTITO Pennac D., "La prosivendola"  
PRESTITO Yourcenar M., "L'opera al nero"  
RESTITUZIONE Bertossi A., "Algoritmi e Strutture Dati"  
RESTITUZIONE Shakespeare W., "Le allegre comari di Windsor"  
FINE

## 2.1.2 Uscita

Il programma deve restituire anzi tutto la lista dei libri ancora sullo scaffale, un libro per riga, inaugurata da una riga che contiene solo la parola chiave

SCAFFALE

e terminata da una riga che contiene solo la parola chiave:

FINE

Quindi, deve fornire una serie di istruzioni per i bibliotecari, inaugurata da una riga che contiene solo la parola chiave

ISTRUZIONI

cui segue un'istruzione per riga, nel formato:

METTERE *autore1*, "*titolo1*" DOPO *autore2*, "*titolo2*"

o, nel caso speciale di un libro che compare al primo posto nella collezione:

METTERE *autore*, "*titolo*" PER PRIMO

Questa serie di istruzioni termina con una riga che contiene solo la parola:

FINE

Infine, il programma deve restituire la lista dei libri che sono sullo scaffale al termine delle istruzioni di riordino, con lo stesso formato descritto in precedenza.

L'uscita corrispondente all'esempio di ingresso su riportato è:

SCAFFALE

Chaucer G., "I Racconti di Canterbury"

Hoeg P., "Il senso di Smilla per la neve"

Hofstadter D., "Godel, Escher, Bach"

King S., "L'ombra dello scorpione"

FINE

ISTRUZIONI

Mettere Shakespeare W., "Le allegre comari di Windsor" dopo King S., "L'ombra dello scorpione"

Mettere Bertossi A., "Algoritmi e Strutture Dati" per primo

Mettere Kernighan B. e Ritchie D., "Il linguaggio C" dopo Hofstadter D., "Godel, Escher, Bach"

FINE

SCAFFALE

Bertossi A., "Algoritmi e Strutture Dati"

Chaucer G., "I Racconti di Canterbury"

Hoeg P., "Il senso di Smilla per la neve"

Hofstadter D., "Godel, Escher, Bach"

Kernighan B. e Ritchie D., "Il linguaggio C"

King S., "L'ombra dello scorpione"

Shakespeare W., "Le allegre comari di Windsor"

FINE

## 2.2 Modello

Le operazioni richieste sono:

- scorrere i libri sullo scaffale per stamparne l'elenco;
- tenere traccia dei libri prestati e dei libri resi;
- fornire informazioni sulla posizione dei libri da rimettere nella libreria;
- determinare la posizione corretta di un libro sullo scaffale in base all'ordine alfabetico.

Occorre quindi gestire tre insiemi di libri: quelli sullo scaffale, quelli in prestito e quelli restituiti. Il primo e il terzo insieme sono ordinati e, in particolare, il terzo consente solo l'accesso all'ultimo libro aggiunto all'insieme (*LIFO = Last In First Out*). Viene quindi spontaneo gestire almeno questi due insiemi come liste e per semplicità gestiremo così anche l'altro. Tutte le operazioni prima elencate verranno quindi realizzate come operazioni su liste.

## 2.3 Codice

### 2.3.1 Prima fase

*Struttura generale del programma e definizione delle strutture dati*

Il file `biblio1.c` fornisce tramite la direttiva `#define` alcune costanti simboliche per:

- la gestione degli errori (codici di uscita come `EXIT_SUCCESS`);
- la dimensione massima per le stringhe di caratteri (`RIGA`, di valore 80).
- i valore logici *vero* (`TRUE`, di valore 1, e `FALSE`, di valore 0);

Questi ultimi vengono sfruttati per definire il tipo enumerativo `enum _boolean`, che per semplicità viene ribattezzato *boolean* tramite il comando `typedef`.

Altri tipi ausiliari di dati specifici dell'esercizio sono:

- *AutoreTitolo*: è una stringa di caratteri, di lunghezza `RIGA`, che contiene autore e titolo di un libro
- *libro*: è una struttura contenente un *AutoreTitolo*

Infine, dato che raccoglieremo i libri in liste, per gestire queste liste viene fornita anche un'interfaccia che consente di eseguire operazioni su liste. Nomi e prototipi delle funzioni di tale libreria di interfaccia derivano dal capitolo 2 del testo di Algoritmi e Strutture Dati di A. Bertossi<sup>1</sup>.

```
listalibri CreaListaLibri ();
void DistruggeListaLibri (listalibri *pL);
boolean listavuota (listalibri L);
poslibro primolista (listalibri L);
poslibro ultimolista (listalibri L);
poslibro succlista (poslibro p, listalibri L);
poslibro preclista (poslibro p, listalibri L);
boolean finelista (poslibro p, listalibri L);
char *leggilista (poslibro p, listalibri L);
listalibri scrivilista (char *AutoreTitolo, poslibro p, listalibri L);
void inslista (char *AutoreTitolo, poslibro p, listalibri L);
void canclista (poslibro *p, listalibri L);
poslibro trovaelemento (char *AutoreTitolo, listalibri L);
void StampaListaLibri (listalibri L);
```

In particolare, sono forniti i tipi ausiliari

- *listalibri* per descrivere una lista
- *poslibro* per descrivere un indice di scorrimento lungo una lista

Poiché la lista viene realizzata come doppia lista a puntatori circolare e con sentinella, entrambi i tipi sono definiti come puntatori a *libro* e la struttura *libro* contiene anche un puntatore al libro successivo (*next*) e uno al libro precedente (*prev*), ovviamente di valore NULL per libri che non appartengono a nessuna lista.

Nel *main* sono definite tre variabili di tipo *listalibri*:

- *Scaffale*, che contiene i libri presenti nella libreria;
- *Prestati*, che contiene i libri in prestito;
- *Resi*, che contiene i libri resi e non riposti.

La struttura del main è la consueta, salvo che si eseguono due algoritmi anziché uno (l'esecuzione dei movimenti dei libri e il reinserimento dei libri resi a fine giornata) e ciascuno di loro è seguito dalla stampa dei suoi risultati.

---

<sup>1</sup>I nomi sono tutti autoesplicativi; l'unica precisazione necessaria è che la funzione *inslista* inserisce il suo primo argomento nella posizione che precede il secondo argomento.

### 2.3.2 Seconda fase

*Interpretazione della linea di comando e caricamento della dotazione della biblioteca*

Questa fase richiede di definire le seguenti funzioni:

- *LeggeLineaComando* che interpreta la linea di comando: somiglia a quella definita nell'esercizio precedente, ma gli argomenti in ingresso sono due: il file dei libri disponibili e quello dei movimenti;
- *CaricaBiblioteca* che legge dal primo file di ingresso una riga alla volta<sup>2</sup> e lo stampa a video finché non legge la terminazione FINE<sup>3</sup>.

### 2.3.3 Terza fase

*Costruzione della lista Scaffale e principio della libreria per gestire liste*

Questa fase richiede di modificare la procedura *CaricaBiblioteca* in modo che le stringhe *AutoreTitolo* lette via via non vengano più stampate a video, ma inserite nella lista *Scaffale*<sup>4</sup>.

Questa fase richiede inoltre di definire alcune le seguenti procedure per la gestione di liste:

- *CreaListaLibri*: crea una lista di libri vuota, cioè contenente la sola sentinella<sup>5</sup>; se la memoria è insufficiente, esce con un messaggio di errore.
- *DistruggeListaLibri*: scorre una lista eliminandone gli elementi uno ad uno finché la lista non è vuota<sup>6</sup>; quindi, dealloca la sentinella.
- *StampaListaLibri*: scorre una lista stampando per ogni libro l'autore e il titolo<sup>7</sup>.

---

<sup>2</sup>Suggerimento: `fgets`, e `strlen` per eliminare l'a capo finale

<sup>3</sup>Suggerimento: `strcmp`

<sup>4</sup>Suggerimento: occorrono le funzioni *inslista* e *ultimolista* (e *succlista*, dato che *inslista* inserisce il primo argomento nella posizione che precede il secondo argomento anziché in quella successiva).

<sup>5</sup>Suggerimento: `malloc`

<sup>6</sup>Suggerimento: *primolista*, *listavuota*, *canclista*

<sup>7</sup>Suggerimento: *primolista*, *finelista*, *succlista*, *leggilista*.

### 2.3.4 Quarta fase

*Conclusione della libreria per gestire liste*

Questa fase richiede di definire le funzioni rimanenti della libreria:

- *primolista*, che restituisce il primo elemento della lista;
- *listavuota*, che restituisce TRUE qualora la lista sia vuota, FALSE altrimenti;
- *canclista*, che cancella l'elemento indicato dalla lista e aggiorna la posizione spostandola sul successivo;
- *finelista* che restituisce TRUE qualora la posizione corrente coincida con quella finale della lista, FALSE altrimenti.
- *leggilista* che restituisce il titolo e l'autore del libro in posizione  $p$  nella lista  $L$ .

### 2.3.5 Quinta fase

*Esecuzione di prestiti e rese dal punto di vista delle liste dei libri prestati e resi, aggiornando anche la lista dei libri disponibili*

Questa fase richiede di realizzare l'algoritmo vero e proprio, gestendo l'esecuzione di prestiti e rese, cioè modificando le corrispondenti liste. Va definita la funzione *EsegueMovimenti* che:

- in caso di prestito:
  1. cerca il libro sullo scaffale, chiamando un'apposita funzione *trovaelemento* (da dichiarare solamente), e lo cancella
  2. inserisce il nuovo libro in coda alla lista dei prestiti
- in caso di resa:
  1. cerca il libro nella lista dei prestiti con la funzione *trovaelemento* e lo cancella
  2. inserisce il libro in cima alla lista dei resi
- in fase di riordino
  1. cerca il libro in cima alla lista dei resi e lo cancella
  2. lo aggiunge nella posizione corretta sullo scaffale
  3. completa l'esecuzione dei prestiti e dei resi aggiornando anche la lista dei libri disponibili

### 2.3.6 Sesta fase

*Reinserimento dei libri resi e definizione di findelemento*

Questa fase richiede di definire le funzioni:

- *trovaelemento* che scorre una lista di libri alla ricerca di un determinato elemento<sup>8</sup>.
- *RiponeLibriRestituiti*, che cerca nella lista *Scaffale* la posizione corretta in base all'ordine alfabetico<sup>9</sup> per ciascun libro della lista *Resi* e lo inserisce in essa.

### 2.3.7 Codice

---

<sup>8</sup>Suggerimento: *primolista*, *finelista*, *succlista*, *leggilista*, `strcmp`

<sup>9</sup>Suggerimento: *primolista*, *finelista*, *succlista*, *leggilista*, `strcmp`