

# DOCUMENTO DI PROGETTO

- **Nome e Cognome:** Lorenzo Bombaci
- **N. matricola:** 515430
- **Progetto:** Gestionale di una biblioteca

## Descrizione:

Il progetto una volta eseguito stampa a video un menù che permette di scegliere tra login e registrazione.

E' stato gestito l'accesso sia come utente sia come amministratore: l'utente, dopo essersi registrato ed aver fatto il login, può prendere in prestito un libro, restituirlo, stampare a video la lista dei libri presenti (disponibili e non), stampare la lista dei libri che ha attualmente preso in prestito, vedere il suo storico (libri restituiti) e scrivere, modificare o leggere le recensioni dei libri.

L'amministratore non ha bisogno di registrarsi, accede inserendo il codice "0000", egli può inserire o cancellare un libro dalla lista, cancellare un utente, stampare la lista degli utenti registrati, vedere la lista libri, lo storico dei libri presi e restituiti e le recensioni di un determinato utente.

Quando un utente stampa la lista libri visualizzerà genere, titolo, autore e codice (che verrà generato automaticamente), l'admin oltre a questi 4 campi visualizzerà anche l'username dell'utente che ha preso in prestito un determinato libro insieme alla data di prenotazione. (se il libro non è stato preso in prestito da nessuno apparirà il simbolo "-")

Mentre l'utente può vedere solo il suo storico, l'amministratore potrà effettuare una ricerca dell'utente di cui vedere lo storico e dopo aver inserito l'username verrà stampato a video lo storico dell'utente richiesto.

## Tipi di dato astratto

Utente: che contiene un campo username, un campo password, due puntatori (head e tail) alla lista dei libri presi in prestito dal cliente e due puntatori (head e tail) alla lista delle recensioni scritte dallo stesso utente.

Nodo\_utente: è un nodo della lista utenti

Libro: contiene i campi genere, titolo, autore, codice, username, data di prenotazione e data di restituzione e due campi puntatori (head e tail) alla lista delle recensioni.

Anche questa lista è una lista di puntatori alle recensioni.

Nodo\_libro: è un nodo della lista libri

Recensione: contiene un campo testo (per scrivere la recensione) un campo username, codice libro e numero di recensione.

Nodo\_recensioni: è un nodo della lista delle recensioni.

Nodo\_punt: è un nodo della lista di puntatori della lista utenti (che puntano alle recensioni di quello stesso utente)

Nodo\_punt\_libro: è un nodo della lista di libri della lista libri (che puntano alle recensioni di quel libro)

# Strutture dati

La struttura dati principale è una lista concatenata realizzata utilizzando i puntatori, oltre al puntatore a head ho utilizzato anche un puntatore a tail per ottimizzare l'inserimento in coda.

Questa lista rappresenta la lista degli utenti registrati (lista utenti).

Le operazioni possibili su questa struttura dati sono:

1. Inserimento utenti
2. Cancellazione utenti
3. Stampa lista utenti
4. Modifica password

Ad ogni utente è associato un username, una password, un puntatore che punta alla lista dei libri che ha attualmente preso in prestito (lista "i miei libri") ed un puntatore che punta alle recensioni scritte dall'utente (recensioni); Queste due liste sono liste di puntatori che puntano rispettivamente ai libri e alle recensioni (strutture dati descritte di seguito); ciò è stato fatto per evitare di duplicare informazioni.

Le operazioni possibili sulla lista "i miei libri" sono:

1. Prendi in prestito un libro
2. Restituisci un libro
3. Stampa lista "i miei libri"

L'operazione possibile sulla lista di recensioni associata all'utente è:

1. Stampa recensioni per utente

Tale struttura dati è dunque una **lista** concatenata di **liste di puntatori**.

Una seconda struttura dati, realizzata come lista concatenata rappresenta la lista dei libri registrati in biblioteca.

Anche in questa lista ho utilizzato sia un puntatore a head sia un puntatore a tail per ottimizzare l'inserimento in coda.

Le operazioni possibili su questa struttura dati sono:

1. Inserimento libri
2. Cancellazione libri
3. Stampa lista libri

Ad ogni libro sono associati i campi: genere, titolo, autore, username, data di prenotazione e reso, e un puntatore che punta alla lista delle recensioni associate al libro. Questa lista è una lista di puntatori che puntano alle recensioni (struttura dati descritte di seguito); ciò è stato fatto per evitare di duplicare informazioni

Anche questa struttura dati è dunque una **lista** concatenata di **liste di puntatori**.

Una terza struttura dati è la lista delle recensioni, ogni utente può scrivere una recensione su un libro o modificarla e visualizzare la lista delle recensioni per ogni libro; anche in questa lista ho utilizzato sia un puntatore a head sia un puntatore a tail per ottimizzare l'inserimento in coda.

Le operazioni possibili sono:

1. Scrivere una recensione
2. Modificare una recensione

Dopo ogni inserimento o cancellazione o modifiche ciascuna lista viene aggiornata e viene salvato su file l'aggiornamento effettuato in modo da poter ricreare la lista ad ogni riavvio del programma; ho implementato anche la funzionalità stampa storico che consiste nel leggere da file l'username a cui è associato un libro e stampare tutti i libri relativi a quell'username, che, a differenza della lista "i miei libri" che stampa solo quelli attualmente in prestito, stampa la lista di tutti i libri presi in prestito e già restituiti dall'utente.

# FUNZIONI

## Utenti.h

**struct nodo\_utente\* trova\_utente(struct nodo\_utente\* head, char username[20]):**

cerca un utente passando come parametro l'username

**struct nodo\_utente\* crealista\_utente(utente profilo, struct nodo\_utente\*\* tail\_utente):**

funzione che crea il primo elemento della lista utenti

**void stampa\_utenti(struct nodo\_utente\* lista\_utenti):**

funzione che permette di stampare a video la lista degli utenti registrati;

**int cancellazione\_utenti(struct nodo\_utente\*\* lista\_utenti, struct nodo\_utente\*\* tail\_utente):**

funzione che permette di cancellare un utente dalla lista utenti;

**int inserimento\_coda (utente account, struct nodo\_utente\*\* lista\_utenti, struct nodo\_utente\*\* tail\_utente):** funzione che inserisce in coda un utente.

**void leggibile(struct nodo\_utente\*\* lista\_utenti, struct nodo\_utente\*\* tail\_utente):**

funzione che legge dal file lista\_utenti.txt

**void scrivibile(struct nodo\_utente\* lista\_utenti):**

funzione che scrive sul file lista\_utenti.txt la lista degli utenti registrati

**int inserimento\_ordinato (utente account, struct nodo\_utente\*\* lista\_utenti, struct nodo\_utente\*\* tail\_utente):** inserisce in ordine alfabetico gli utenti che via via si registrano

**void registrazione(struct nodo\_utente\*\* lista\_utenti, struct nodo\_utente\*\* tail\_utente);**

funzione che permette all'utente di registrarsi;

**bool trova(utente account, struct nodo\_utente\* lista\_utenti):**

funzione che controlla (al momento del login) se gli elementi inseriti esistono, in caso affermativo stampa il menu;

**void modifica\_password(struct nodo\_utente\* lista\_utenti):**

funzione che permette all'utente di modificare la sua password modificando sul file il campo password dell'utente

**void stampa\_storico(char username[20]):**

funzione che stampa a video lo storico di un determinato utente;

**int cerca\_storico\_utente (struct nodo\_utente\* lista\_utenti):**

funzione che permette all'admin di cercare lo storico di un determinato utente

# Libri.h

**struct nodo\_libro\* trova\_libro(struct nodo\_libro\* head, char codice[10]):**

funzione che con un ciclo while scandisce la lista libri e trova il libro

**void stampa\_libri(bool admin, struct nodo\_libro\* lista\_libri):**

stampa richiesto la lista dei libri, questa funzione è disponibile sia per utente che per admin, la differenza è che l'amministratore oltre a vedere genere titolo autore e codice (come l'utente) vedrà anche il campo username e la eventuale data di prenotazione del libro

**int cancellazione\_libro(struct nodo\_libro\*\* lista\_libri, struct nodo\_libro\*\* tail\_libro):**

funzione che cancella un libro dalla lista libri

**int cancellazione\_my\_libri(struct nodo\_punt\_libro\*\* head, char codice[10]):** funzione che cancella un libro dalla lista dei libri in possesso da un determinato utente;

**int inserimento\_coda\_libri (struct nodo\_libro\*\* lista\_libri, struct nodo\_libro\*\* tail\_libro, libro book):** funzione che inserisce un libro in coda alla lista

**void scrivi\_my\_libri(struct nodo\_libro\* indirizzo, char username[20], struct nodo\_utente\* lista\_utenti):** funzione che crea la lista dei libri in possesso da un determinato utente;

**void stampa\_my\_libri(char username[20], struct nodo\_utente\* lista\_utenti):**

funzione che stampa la lista dei libri prenotati da un determinato utente

**void leggi\_file\_libro(struct nodo\_libro\*\* lista\_libri, struct nodo\_libro\*\* tail\_libro, struct nodo\_utente\* lista\_utenti):** funzione che legge il file "lista\_libri.txt";

**void scrivi\_file\_libro(struct nodo\_libro\* lista\_libri):** funzione che aggiorna il file "lista\_libri.txt";

**void registrazione\_libri(struct nodo\_libro\*\* lista\_libri, struct nodo\_libro\*\* tail\_libro):**

funzione che permette all'admin di inserire un nuovo libro in lista;

**void prendi\_libro(utente account, struct nodo\_libro\* lista\_libri, struct nodo\_utente\* lista\_utenti):**

funzione per prendere in prestito un libro;

**void scrivi\_file\_storico(libro book):** funzione che scrive sul file "storico.txt";

**void restituisci\_libro(utente account, char username[20], struct nodo\_libro\* lista\_libri, struct nodo\_utente\* lista\_utenti):** funzione per prendere in prestito un libro;

**void modifica(char codice[10], struct nodo\_libro\* lista\_libri):**

Funzione necessaria per la restituzione di un libro: quando un libro viene preso in prestito il campo username (prima vuoto) verrà occupato dall'username dell'utente che ha preso il libro. Al momento della restituzione questa funzione modifica il campo username con un campo vuoto in modo da rendere nuovamente disponibile il libro

# Recensioni.h

**int inserimento\_coda\_puntatori (struct nodo\_punt\*\* head, struct nodo\_punt\*\* tail, struct nodo\_recensione\* indirizzo):**

funzione che inserisce in coda alla lista di puntatori della lista utenti un puntatore alle recensioni

**int inserimento\_coda\_punt\_libri (struct nodo\_punt\_libro\*\* head, struct nodo\_punt\_libro\*\* tail, struct nodo\_libro\* indirizzo):**

funzione che inserisce in coda alla lista di puntatori della lista libri un puntatore alle recensioni

**void leggi\_file\_recensione (struct nodo\_recensione\*\* lista\_recensioni,**

**struct nodo\_recensione\*\* tail\_recensione, struct nodo\_libro\* lista\_libri, struct nodo\_utente\* lista\_utenti):**

funzione che legge dal file recensioni.txt

**void stampa\_recensioni (struct nodo\_libro\* head\_l, struct nodo\_utente\* head\_u, char username[20], bool admin):**

funzione che stampa le recensioni per utente o per libro; nel caso dell'utente si potrà stampare la lista delle proprie recensioni, nel caso dell'admin inserendo l'username dell'utente di cui vedere le recensioni si potranno leggere le recensioni di un qualsiasi utente.

**int inserimento\_coda\_recensioni (struct nodo\_recensione\*\* head, struct nodo\_recensione\*\* tail, recensione\* recensione):**

funzione che inserisce in coda alla lista la recensione appena inserita

**int scrivi\_recensione (struct nodo\_recensione\*\* head, struct nodo\_recensione\*\* tail, char username[20], struct nodo\_libro\* lista\_libri):**

funzione che registra una recensione sulla lista recensioni

**void scrivi\_file\_recensioni(struct nodo\_recensione\* head);**

funzione che aggiorna il file recensioni.txt

**void modifica\_recensione(struct nodo\_recensione\* lista\_recensioni, char username[20]);**

funzione che permette all'utente di modificare una recensione che ha già scritto in passato

**void login(struct nodo\_utente\* lista\_utenti, struct nodo\_utente\* tail\_utente,**

**struct nodo\_libro\* lista\_libri, struct nodo\_libro\* tail\_libro,**

**struct nodo\_recensione\* lista\_recensioni, struct nodo\_recensione\* tail\_recensione):**

funzione che permette all'utente di effettuare il login

# File utenti.h

```
1
2 typedef struct {
3     char username[20];
4     char password[20];
5     struct nodo_punt_libro* my_book; //faccio una lista di liste
6     struct nodo_punt_libro* my_book_tail; //
7     struct nodo_punt* my_recensioni; //lista di liste
8     struct nodo_punt* my_recensioni_tail;
9 } utente;
10
11 struct nodo_utente{
12     utente account;
13     struct nodo_utente* next;
14 };
15 struct nodo_utente* trova_utente(struct nodo_utente* head, char username[20]);
16
17 //funzione che crea una lista UTENTI dal primo elemento
18 struct nodo_utente* crealista_utente(utente profilo, struct nodo_utente** tail_utente);
19
20 // funzione stampa UTENTI
21 void stampa_utenti(struct nodo_utente* lista_utenti);
22
23 //scrittura su file lista_utenti.txt
24 void scrivifile(struct nodo_utente* lista_utenti);
25
26 //funzione cancellazione UTENTI
27 int cancellazione_utenti(struct nodo_utente** lista_utenti, struct nodo_utente** tail_utente);
28
29 // inserimento in coda UTENTI
30 int inserimento_coda (utente account, struct nodo_utente** lista_utenti, struct nodo_utente** tail_utente);
31
32 //leggere il file lista_utenti.txt
33 void leggiifile(struct nodo_utente** lista_utenti, struct nodo_utente** tail_utente);
34
35 //inserisce un elemento ordinatamente in lista
36 int inserimento_ordinato (utente account, struct nodo_utente** lista_utenti, struct nodo_utente** tail_utente);
37
38 bool trova(utente account, struct nodo_utente* lista_utenti);
39
40 //funzione registrazione UTENTI
41 void registrazione(struct nodo_utente** lista_utenti, struct nodo_utente** tail_utente);
42
43 void modifica_password(struct nodo_utente* lista_utenti);
44
45 //funzione per stampare lo storico utente
46 void stampa_storico(char username[20]);
47
48 //controllo username e controllo data
49 int cerca_storico_utente (struct nodo_utente* lista_utenti);
50
```

# File libri.h

```
1 typedef struct {
2     char titolo[40];
3     char autore[40];
4     char genere[40];
5     char codice[10];
6     char data_prenotazione[20];
7     char data_restituzione[20];
8     char username[20];
9     struct nodo_punt* my_recensioni;
10    struct nodo_punt* my_recensioni_tail;
11 } libro;
12
13 struct nodo_libro{
14     libro book;
15     struct nodo_libro* next;
16 };
17 //funzione che trova un libro passando il codice come parametro
18 struct nodo_libro* trova_libro(struct nodo_libro* head, char codice[10]);
19 // funzione che crea una lista LIBRI dal primo elemento
20 struct nodo_libro* crealista_libro(libro dati);
21 // funzione stampa LIBRI
22 void stampa_libri(bool admin, struct nodo_libro* lista_libri);
23 //funzione cancellazione LIBRI
24 int cancellazione_libro(struct nodo_libro** lista_libri, struct nodo_libro** tail_libro);
25 //funzione cancellazione miei libri
26 int cancellazione_my_libri(struct nodo_punt_libro** head, char codice[10]);
27 //inserimento in coda LIBRI
28 int inserimento_coda_libri (struct nodo_libro** lista_libri, struct nodo_libro** tail_libro, libro book);
29 //scrive la lista dei libri presi in prestito da un utente
30 void scrivi_my_libri(struct nodo_libro* indirizzo, char username[20], struct nodo_utente* lista_utenti);
31 //stampa la lista dei libri presi in prestito da un utente
32 void stampa_my_libri(char username[20], struct nodo_utente* lista_utenti);
33 //leggere il file lista_libri.txt
34 void leggi_file_libro(struct nodo_libro** lista_libri, struct nodo_libro** tail_libro, struct nodo_utente* lista_utenti);
35
36 //scrivi su file lista_libri.txt
37 void scrivifile_libro(struct nodo_libro* lista_libri);
38
39 //funzione registrazione LIBRI
40 void registrazione_libri(struct nodo_libro** lista_libri, struct nodo_libro** tail_libro);
41
42 // prendi un libro
43 void prendi_libro(utente account, struct nodo_libro* lista_libri, struct nodo_utente* lista_utenti);
44
45 //funzione che modifica i campi username e data al momento della restituzione
46 void modifica(char codice[10], struct nodo_libro* lista_libri);
47
48 //funzione per restituire un libro e che lo rende nuovamente disponibile
49 void restituisci_libro(utente account, char username[20], struct nodo_libro* lista_libri, struct nodo_utente* lista_utenti);
50
51 //scrivi su file lista_libri.txt
52 void scrivifile_storico(libro book);
53
```



# File recensioni.h

```
1 #include "utenti.h"
2 #include "libri.h"
3
4 typedef struct {
5     char testo [200];
6     char username[20];
7     char codice[10];
8     char numero[10];
9 } recensioni;
10
11 struct nodo_recensione{
12     recensioni recensione;
13     struct nodo_recensione* next;
14 };
15
16 struct nodo_punt{
17     struct nodo_recensione* indirizzo; //lista di puntatori ovvero il campo dati è un puntatore
18     struct nodo_punt* next;
19 };
20
21 struct nodo_punt_libro{
22     struct nodo_libro* indirizzo;
23     struct nodo_punt_libro* next;
24 };
25
26 //inserimento in coda di un puntatore nella lista di puntatori della lista utenti
27 int inserimento_coda_puntatori (struct nodo_punt** head, struct nodo_punt** tail, struct nodo_recensione* indirizzo);
28
29 //inserimento in coda di un puntatore nella lista di puntatori della lista libri
30 int inserimento_coda_punt_libri (struct nodo_punt_libro** head, struct nodo_punt_libro** tail, struct nodo_libro* indirizzo);
31
32 //leggere il file lista_recensioni.txt
33 void leggi_file_recensione(struct nodo_recensione** lista_recensioni, struct nodo_recensione** tail_recensione,
34     struct nodo_libro* lista_libri, struct nodo_utente* lista_utenti);
35
36 // funzione stampa RECENSIONI
37 void stampa_recensioni (struct nodo_libro* head_l, struct nodo_utente* head_u, char username[20], bool admin);
38
39 //inserimento in coda RECENSIONI
40 int inserimento_coda_recensioni (struct nodo_recensione** head, struct nodo_recensione** tail, recensioni recensione);
41
42 //funzione per scrivere una recensione
43 int scrivi_recensione (struct nodo_recensione** head, struct nodo_recensione** tail, char username[20], struct nodo_libro* lista_libri);
44
45 //scrittura su file lista_recensioni.txt
46 void scrivi_file_recensioni(struct nodo_recensione* head);
47
48 //modifica il testo di una recensione
49 void modifica_recensione(struct nodo_recensione* lista_recensioni, char username[20]);
50
51 void login(struct nodo_utente* lista_utenti, struct nodo_utente* tail_utente, struct nodo_libro* lista_libri, struct nodo_libro* tail_libro,
52     struct nodo_recensione* lista_recensioni, struct nodo_recensione* tail_recensione);
53
```

# Interfaccia grafica

2 richiama la funzione registrazione, crea una lista utenti che vengono inseriti in ordine alfabetico con la funzione "inserimento\_ordinato"

```
inserire username:lorenzo
inserire password:1234
```

```
username=lorenzo
password=1234
```

```
REGISTRAZIONE EFFETTUATA
```

```
BIBLIOTECA

MENU PRINCIPALE
1 login
2 registrazione
3 exit
```

1 richiama la funzione login

```
1 login utente
2 login amministratore
3 exit
```

2 richiama la funzione "restituisce\_libro" e chiamando all'interno la funzione "modifica", sostituisce il campo username e data prenotazione con campi vuoti

```
1
cerca un libro per:
1 genere
2 titolo
3 autore
4 annulla
```

3 ritorna al menù principale

```
MENU' UTENTE
1 PRENDI IN PRESTITO UN LIBRO
2 RESTITUISCI LIBRO
3 VISUALIZZA LISTA LIBRI
4 I MIEI LIBRI
5 STORICO
6 MODIFICA PASSWORD
7 SCRIVI UNA RECENSIONE
8 MODIFICA RECENSIONE
9 CERCA RECENSIONI
10 exit
```

Richiama la funzione "prendi\_libro": verrà stampato l'elenco dei libri con i relativi codici trovati in base alla ricerca per genere, titolo o autore (1,2,3). Scegliendo 4 invece si ritorna al menù utente.

```
inserisci codice del libro che vuoi restituire:
```

```
1
codice ricevutolibro trovato
```

```
fantasy
```

```
Harry Potter
```

```
J.K. Rowling
```

```
1
```

```
29/08/2021
```

```
confermi la tua scelta?
```

```
1 SI
```

```
2 NO
```

```
1
```

```
inserisci data odierna
```

```
30/08/2021
```

```
libro restituito
```

2 torna al menu

```
inserisci codice del libro che vuoi prendere in prestito
```

```
1
confermi la tua scelta?
```

```
1 SI
```

```
2 NO
```

```
1
```

```
inserisci data
```

```
29/08/2021
```

```
il libro e' stato preso in prestito
```

2 torna al menu

```
fantasy
Harry Potter
J.K. Rowling
1
```

Richiama la funzione "stampa\_libri" e stampa l'elenco di

TUTTI i libri presenti in biblioteca

(per semplicità di rappresentazione supponiamo che al momento sia inserito solamente un libro)

```
fantasy
Harry Potter
J.K. Rowling
1
29/08/2021
30/08/2021
lorenzo
```

```
I MIEI LIBRI
fantasy
Harry Potter
J.K. Rowling
1
29/08/2021
```

Richiama la funzione "stampa\_my\_libri" e stampa l'elenco dei libri presi in prestito.

Richiama la funzione "stampa\_storico" e stampa lo storico dell'utente.

Ogni utente può vedere solo il suo storico, lo storico si aggiorna solo dopo aver restituito il libro

Richiama la funzione  
"modifica\_password", agisce  
sulla lista utenti e modifica il  
campo password con i nuovi  
caratteri inseriti, dopo ritorna

```
2
inserisci codice del libro
1
NUMERO RECENSIONE: 1
UTENTE: lorenzo
LIBRO: 1
RECENSIONE: *NUOVO TESTO*
```

```
1
NUMERO RECENSIONE: 1
UTENTE: lorenzo
LIBRO: 1
RECENSIONE: *NUOVO TESTO*
```

```
6
inserisci password attuale
1234
inserisci nuova password
5678
```

PASSWORD MODIFICATA!

```
1 LE MIE RECENSIONI
2 RECENSIONE PER LIBRO
```

```
MENU' UTENTE
1 PRENDI IN PRESTITO UN LIBRO
2 RESTITUISCI LIBRO
3 VISUALIZZA LISTA LIBRI
4 I MIEI LIBRI
5 STORICO
6 MODIFICA PASSWORD
7 SCRIVI UNA RECENSIONE
8 MODIFICA RECENSIONE
9 CERCA RECENSIONI
10 exit
```

Scegliendo 10 si ritorna al  
menù principale

```
7
inserisci codice del libro che vuoi recensire:
1
scrivi una recensione
*TESTO*
NUMERO RECENSIONE: 1
```

Permette di scrivere il campo testo della recen-  
sione, dopo aver fatto genererà un numero del-  
la recensione

Richiama la funzione "scrivi\_recensione" inoltre  
utilizza 2 volte la funzione "inserimento\_coda  
puntatori" (una volta per il libro e una volta  
per l'utente perché ogni recensione è associata  
sia ad un libro sia all'utente).

```
8
inserisci numero di recensione
1
inserisci nuova recensione
*NUOVO TESTO*
recensione modificata
```

Richiama la funzione  
"modifica\_recensione", una volta  
inserite le modifiche viene modifi-  
cato il campo testo della recensio-  
ne e dopo ritorna al menù.

richiama la funzione "stampa\_recensioni", ci  
sono 2 diverse possibilità:

Con un ciclo while si scandisce la lista e in base  
alla scelta stampa l'elenco delle recensioni ri-  
chieste:

**1** con il ciclo while si cercano tutte le recensioni  
scritte da un determinato username (in questo  
caso dell'utente con cui si è effettuato l'accesso)  
e vengono stampate a video

**2** con il ciclo while si cercano e si stampano  
tutte le recensioni di un determinato libro (per  
esempio il codice "1") e vengono stampate a  
video

```
BIBLIOTECA

MENU PRINCIPALE
1 login      2 registrazione
3 exit
```

3 esce dal programma

```
1 login utente
2 login amministratore
```

A differenza dell'utente, l'admin con la funzione stampa\_libri può vedere (oltre a genere, titolo, autore e codice) anche la data di prenotazione e l'utente che ha preso in prestito il libro (se il campo è vuoto apparirà il simbolo "-")

```
1
fantasy
Harry Potter
J.K. Rowling
1
-
```

```
MENU' ADMIN

1 STAMPA LISTA LIBRI
2 INSERISCI LIBRO
3 CANCELLA LIBRO
4 CANCELLA UTENTE
5 STAMPA LISTA UTENTI
6 STAMPA STORICO
7 STAMPA RECENSIONI
8 exit
```

L'admin non ha bisogno di registrarsi, entrerà con un codice di accesso che conosce solo lui (il codice è 0000)

```
scrivi utente da cancellare:
lorenzo
l'elemento e' stato cancellato
```

Richiama la "funzione cancellazione" e funziona allo stesso modo della funzione "cancellazione\_libri"

```
scrivi codice del libro da cancellare:
1
il libro e' stato cancellato
```

Richiama la funzione "cancellazione\_libro", si effettuerà una ricerca fino a quando non si troverà un libro con lo stesso codice di quello inserito, una volta cancellato il libro si ritorna al menù

```
inserire genere:
fantasy
inserire titolo:
Harry Potter
inserire autore:
J.K. Rowling
genere=fantasy titolo=Harry Potter autore=J.K. Rowling codice=1
REGISTRAZIONE EFFETTUATA
```

Richiama la funzione “cerca\_storico\_utente”, l’utente può vedere solo il proprio storico, invece l’admin con una ricerca può visualizzare lo storico di qualsiasi utente; dopo aver effettuato la ricerca chiamando la funzione “stampa\_storico” verrà stampato a video lo storico dell’utente cercato

```
inserisci username dell'utente di cui vedere lo storico
lorenzo
utente trovato
fantasy
Harry Potter
J.K. Rowling
1
29/23/2021
30/08/2021
lorenzo
```

```
lorenzo
1234
```

Richiama la funzione “stampa\_utenti” che stampa una lista di tutti gli utenti registrati alla biblioteca (supponiamo che al momento ci sia un solo utente registrato)

5

Scegliendo 8 si ritorna al menù principale

```
MENU' ADMIN
1 STAMPA LISTA LIBRI
2 INSERISCI LIBRO
3 CANCELLA LIBRO
4 CANCELLA UTENTE
5 STAMPA LISTA UTENTI
6 STAMPA STORICO
7 STAMPA RECENSIONI
8 exit
```

6

richiama la funzione “stampa\_recensioni”, ci sono 2 diverse possibilità:

Con un ciclo while si scandisce la lista e in base alla scelta (1 o 2) stampa l’elenco delle recensioni richieste

7

```
1 STAMPA RECENSIONI PER UTENTE
2 RECENSIONE PER LIBRO
```

```
2
inserisci codice del libro
1
NUMERO RECENSIONE: 1
UTENTE: lorenzo
LIBRO: 1
RECENSIONE: *NUOVO TESTO*
```

**2** con il ciclo while si cercano e tutte le recensioni di un determinato libro (per esempio il codice “1”) e vengono stampate a video

```
1
inserisci utente di cui stampare le recensioni
lorenzo
```

**1** con il ciclo while si cercano tutte le recensioni scritte da un determinato username.

In questo caso non abbiamo effettuato un accesso con un utente: l’admin può stampare le recensioni di qualsiasi utente semplicemente inserendo l’username

```
NUMERO RECENSIONE: 1
UTENTE: lorenzo
LIBRO: 1
RECENSIONE: *NUOVO TESTO*
```

Alla fine le recensioni vengono stampate a video e subito dopo si ritorna al menù