***RELAZIONE PROGETTAZIONE LIBRERIA***

***Richieste:***

Il programma gestisce una libreria di libri, consentendo di caricare informazioni da un file CSV, visualizzare libri per categoria e cercare un libro per titolo. Ogni libro è identificato da un titolo, un autore, un anno di pubblicazione, un prezzo e una categoria. Le categorie di libri sono assegnate automaticamente in base ai dati nel file CSV.

### **Strutture e Tipi Dati**

#### ***1. Categoria (Enum)***

Un enumeratore che rappresenta le diverse categorie di libri.

typedef enum {

NARRATIVA,

SAGGISTICA,

SCIENZE,

ARTE,

NUM\_CATEGORIE

} Categoria;

#### ***2. Libro (Struct)***

Una struttura che rappresenta un libro, contenente informazioni su titolo, autore, anno di pubblicazione, prezzo e categoria.

typedef struct {

char titolo[MAX\_SIZE];

char autore[MAX\_SIZE];

int anno\_pubblicazione;

double prezzo;

Categoria categoria;

} Libro;

### ***Funzioni***

#### 1**. stampaLibro**

Stampa i dettagli di un libro.

void stampaLibro(const Libro\* libro);

Parametri:

* libro: puntatore a una struttura Libro.

Funzionalità: Stampa il titolo, l'autore, l'anno di pubblicazione, il prezzo e la categoria del libro.

#### 2. **determineCategoria**

Determina la categoria di un libro basata sul titolo.

Categoria determinaCategoria(const char\* titolo);

Parametri:

* titolo: stringa contenente il titolo del libro.

Funzionalità: Restituisce la categoria appropriata in base al contenuto del titolo. Le categorie possono essere: Narrativa, Saggistica, Scienze, Arte.

#### 3. **caricaLibriDaCSV**

Carica i libri da un file CSV.

int caricaLibriDaCSV(Libro libreria[], const char\* nomeFile);

Parametri:

* libreria[]: array di strutture Libro in cui memorizzare i libri.
* nomeFile: stringa contenente il nome del file CSV da cui caricare i dati.

Ritorna:

* Il numero di libri caricati dal file.

Funzionalità: Apre il file CSV, legge i dati riga per riga e popola l'array di libri. Salta l'intestazione del file CSV e utilizza un ciclo per leggere i dati di ogni libro, suddividendo i dati sulla base delle virgole. Ogni libro viene memorizzato nell'array e la sua categoria viene determinata.

#### 4. **visualizzaLibriPerCategoria**

Visualizza tutti i libri appartenenti a una specifica categoria.

void visualizzaLibriPerCategoria(Libro libreria[], int num\_libri, Categoria categoria);

Parametri:

* libreria[]: array di strutture Libro.
* num\_libri: numero di libri nell'array.
* categoria: categoria da visualizzare.

Funzionalità: Itera attraverso l'array di libri e stampa i dettagli di quelli che appartengono alla categoria specificata.

#### 5. **cercaLibroPerTitolo**

Cerca un libro in base al titolo.

void cercaLibroPerTitolo(Libro libreria[], int num\_libri, const char\* titolo);

Parametri:

* libreria[]: array di strutture Libro.
* num\_libri: numero di libri nell'array.
* titolo: titolo del libro da cercare.

Funzionalità: Cerca nell'array un libro che corrisponda al titolo fornito e stampa i dettagli del libro trovato. Se non viene trovato alcun libro, viene visualizzato un messaggio di errore.

#### 6. **mostraMenu**

Visualizza il menu delle opzioni disponibili per l'utente.

void mostraMenu();

Funzionalità: Stampa le opzioni disponibili nel menu principale del programma.

7. **void trim(char\* str)**

Funzione per rimuovere spazi bianchi iniziali e finali

### **Funzione main**

La funzione main è il punto di ingresso del programma. Qui vengono eseguite le seguenti operazioni:

1. Caricamento dei Libri: Invoca caricaLibriDaCSV per caricare i libri dal file CSV.
2. Interazione con l'Utente: Presenta un menu all'utente e gestisce le scelte attraverso un ciclo do-while:
   * Visualizza libri per categoria.
   * Cerca un libro per titolo.
   * Esce dal programma.