



UNIVERSITÀ  
DI TRENTO

**Corso di Calcolo Numerico e Programmazione**  
*Dipartimento di Ingegneria Industriale*

Pierfrancesco Oselin MAT. 202199

## 1 Introduzione

L'obiettivo di questo progetto è approfondire e implementare la teoria affrontata a lezione in modo da renderla propria. In particolare, lo scopo assegnato è quello di creare un sistema che consenta la gestione di un inventario di libri. Il programma è costituito da un file principale (*main.c*) e da una libreria contenente operazioni di gestione di libri (*booksystem.c*, *booksystem.h*).

## 2 Funzioni implementate

Seguono ora brevi spiegazioni di alcune delle funzioni implementate nel programma.

### 2.1 struct Book

Questa struttura costituisce un vero e proprio libro: contiene infatti titolo, autore, casa editrice e anno di pubblicazione.

### 2.2 struct Library

Questa struttura è una libreria vera e propria. Contiene un intero che fornisce il numero di libri salvati, e un vettore di libri riallocato dinamicamente.

## 2.3 struct Wrap

Questa struttura è utilizzata per le operazioni di sort. Permette di tener traccia del libro una volta convertito in un intero per l'ordinamento.

## 2.4 main

Funzione fondamentale per l'avvio del programma, è composta sostanzialmente da un *if*: se non sono stati forniti parametri da terminale viene avviato il mainloop, altrimenti analizza i parametri forniti e in caso procede con l'elaborazione dati. Viene inoltre attivato il controllo di interruzione da terminale.

## 2.5 mainloop

Ciclo principale e cuore fondamentale del programma, esso è costituito da un ciclo *do-while* attraverso il quale la macchina si pone in attesa di input da parte dell'utente. I comandi sono letti richiamando la funzione *scanf*.

## 2.6 quit

Questa funzione impedisce all'utente di terminare il programma da tastiera (*ctrl-C*) in caso abbia apportato modifiche non salvate, come ad esempio l'aggiunta o la rimozione di un libro. È richiamata dalla funzione *signal*.

## 2.7 merge, mergesort, sort\_ricorsivo e sortdata

Insieme di funzioni che permettono di ordinare il catalogo di libri inseriti.

**sortdata** è la prima funzione ad essere richiamata: imposta le variabili da utilizzare e controlla che la libreria non sia vuota.

**sort\_ricorsivo** Inizialmente non presente, è stata aggiunto per eseguire più operazioni di *sort* in modo da ordinare non solo la prima lettera ma parole intere. In caso di iniziale uguale, è infatti necessario considerare la lettera successiva: dopo un tentativo di ordinamento, se sono presenti casi dubbi, verrà nuovamente applicato il sort solo nell'intervallo compromesso.

**mergesort e merge** Questa funzione costituisce il vero algoritmo del *merge sort*, attraverso il quale l'array da ordinare viene scomposto in array più piccoli fino ad arrivare a vettori di lunghezza 1. Successivamente viene richiamata la funzione *merge*, che permette di riunire tutti gli array, ordinandoli.

## 2.8 getnumber

Operazione fondamentale che trasforma un vettore di parole in un vettore di interi da ordinare.

## 2.9 addbook

Funzione che permette l'aggiunta di un libro all'interno della libreria. Se è stata richiamata dal ciclo principale caricherà il libro in memoria, altrimenti lo aggiungerà direttamente al file di salvataggio.

## 2.10 removebook

Analogo ad *addbook*, con questa funzione viene eliminato un libro dalla libreria se presente. Se la funzione è stata chiamata dal ciclo principale verrà cancellato dalla memoria, altrimenti dal file di salvataggio.

## 2.11 findbook

Questa funzione elenca tutti i libri presenti in libreria. La ricerca avviene per autore.

## 2.12 savedata

Funzione che permette il salvataggio dei dati su file, aggiungendo in coda o sovrascrivendo. A seconda degli input che riceve esegue operazioni diverse.

## 2.13 fetchdata

Funzione che permette il caricamento in memoria di una libreria salvata su file. Anch'essa riceve in input la modalità di funzionamento ed è utilizzata in più funzioni del programma, quali ad esempio *sort*, *find*, etc.