

Guida all'uso della libreria (Corso di Programmazione)

February 2023

1 Introduzione

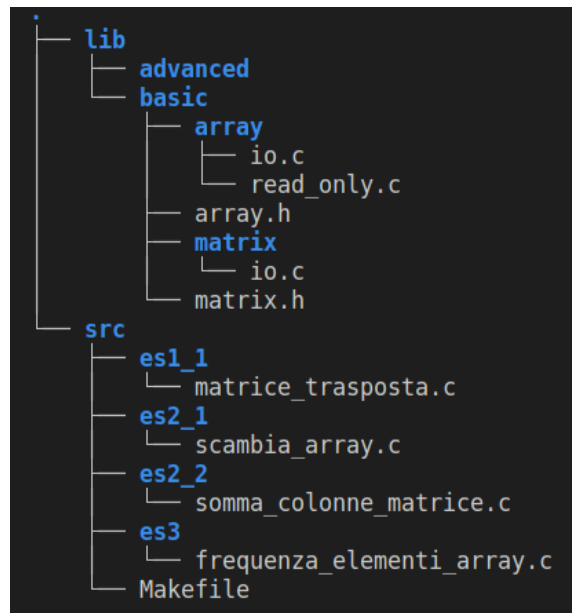
Durante il corso, gli studenti dovranno realizzare, mantenere e aggiornare una libreria C contenente funzioni, macro ecc. utili per la risoluzione di altri esercizi. la libreria è divisa in due parti: *basic*, contenete le funzioni su array, matrici e stringhe (prima parte del corso); e *advanced* contenente funzioni per le strutture dati (seconda parte del corso).

2 Struttura working directory

La libreria, sia per la parte *basic* che *advanced*, è strutturata su due livelli: struttura dati di riferimento (e.g. array) e tipologia di funzionalità implementata (e.g. I/O). Tutte le funzioni di una stessa tipologia verranno implementate in uno stesso file *.c* e tutti i file riferiti ad una struttura saranno raggruppati in una cartella e associati ad un unico header file.

Esempio Una funzione di input/output su array è implementata nel file *basic/array/io.c* ed esposta nel file header *basic/array.h*.

La struttura consigliata per la working directory di un progetto è quindi simile alla seguente (starter code per gli esercizi del Lab 4):



Nella cartella *src* sono contenute le cartelle degli esercizi da svolgere, insieme al *Makefile* (vedi sezione Compilazione).

3 Utilizzo funzioni di libreria

Per utilizzare una funzione di libreria è sufficiente usare la seguente direttiva:

```
#include<*sezione*/*nome_struttura*.h>
```

Ad esempio, per usare una funzione di I/O su array, basterà usare:

```
#include<basic/array.h>
```

4 Compilazione

Nella pagina del corso, nella sezione *Libreria*, troverete lo starter code aggiornato con il Makefile per compilare i diversi esercizi, la libreria ed effettuare il linking. Questo file richiederà modifiche minime durante il corso (potenzialmente nessuna).

Sono disponibili i targets *clean* e un target corrispondente al nome di ogni esercizio (e.g., *es2_1*). Il target *clean* elimina tutti i file oggetto e gli eseguibili. Usando il comando per compilare esercizi verrà creato un eseguibile dal nome corrispondente (e.g., *es2_1.out*). Il tutto è estendibile con nuovi esercizi, sarà sufficiente creare una cartella col nome desiderato.

In caso la struttura della vostra working directory non sia conforme a quanto prescritto (i.e. avete la libreria in un percorso diverso), sarà sufficiente modificare la variabile *LIBS* nel makefile e sostituirla con il percorso della cartella *lib*.

5 Come aggiungere nuove funzionalità

Per aggiungere una nuova funzione di libreria è necessario dichiararne la firma nel corrispondente file header e implementarla in uno qualsiasi dei file c nella sottocartella corrispondente (utilizziamo più file c per una questione di organizzazione).

Ad esempio, se vogliamo implementare la funzione che stampa il primo elemento di un array di interi aggiungeremo le parti in verde ai file *array.h* e *array/io.c*:

```
//array.h
void stampa_array(int*, int);
void leggi_array(int*, int);
int primo_elem(int*, int);

//array/io.c
#include "../array.h"
void stampa_array(int* A, int dim){
    //...
}
void leggi_array(int* A, int dim){
    //...
}
int primo_elem(int* A , int dim){if(dim != 0) return A[0];}
```

Ricompilando la libreria (make ricompilerà solo i file modificati e i target che dipendono da esso), potremo utilizzare la nuova funzione.