

# LIBRERIA

Definizione: una LIBRERIA è, ad ALTO LIVELLO,

- una collezione di funzioni/procedure
- tipicamente accomunate da qualcosa:
  - tipologie operazioni
  - descrizione e manipolazione di entità
- utilizzabile da diversi programmi
- utilizzabile, previa compilazione, in diversi ambienti

Una LIBRERIA NON è un programma!

↳ NON HA UNA FUNZIONE MAIN

Avete già usato librerie: `iostream`  
`fstream`  
`cmath`  
...

## CREAZIONE LIBRERIA

Una libreria è costituita da:

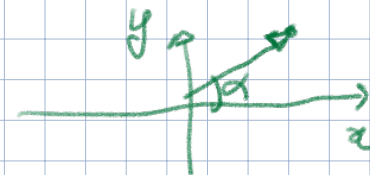
- File HEADER: contiene:
  - dichiarazione dei tipi di dato non di sistema usati/definiti dalla libreria
  - dichiarazione delle funzioni/procedure della libreria
- File Sorgente: contiene le definizioni delle funzioni della libreria
- File OGGETTO: contiene i record di attivazione di ogni funzione/procedure definiti nelle librerie generati dal compilatore in compilazione

## ESEMPIO: LIBRERIA VETTORI

Questa libreria contiene funzionalità per fare operazioni tra vettori in  $\mathbb{R}^2$

### OPERAZIONI:

- somma  $\vec{V}_1 + \vec{V}_2$
- sottrazione  $\vec{V}_1 - \vec{V}_2$
- prodotto scalare  $\vec{V}_1 \cdot \vec{V}_2$
- modulo  $|\vec{V}|$
- angolo (rispetto all'asse)
- componente  $x$  di  $|\vec{V}|$  e  $\alpha$   
y di  $|\vec{V}|$  e  $\alpha$
- test ortogonalità
- ...



- leggi\_textiera  $\vec{V}$
- leggi\_file  $\vec{V}$
- stampa\_video  $\vec{V}$
- stampa\_file  $\vec{V}$

HEADER: vettoriR2.h ← estensione  
nome di libreria

#include ... di tutte le librerie che servono:

- se la libreria che stiamo creando dovrà fare operazioni trigonometriche etc... <cmath>
  - se stampa/legge video/tastiera <iostream>
  - " " " file <fstream>
- ... using namespace std;

#define ... : eventuali costanti utili

ESEMPLO: DIM 2

perché lavoriamo con array di 2 elementi

Poi: tutte le DICHIARAZIONI delle FUNZIONI / PROCEDURE della LIBRERIA

`/x res = v1 + v2 */`

`void somma (double v1[], double v2[], double res[])`

! Documenta la funzione, spiegando il significato dei parametri

Una funzione può restituire 1 valore di un certo tipo  
⇒ non può restituire una dupla di valori...  
... usiamo parametro array per ESPORTARE il risultato

`void sottr ( ... )`

`double prodScalare (double v1[], double v2[]);`

↳ il prodotto scalare  $\vec{v}_1 \cdot \vec{v}_2$  è un valore

...

FILE Sorgente : vettoriR2.c ← estensione  
nome LIBRERIA

#include "vettoriR2.h"

Ovvero, copia/incolla tutto il contenuto di vettoriR2.h

⇒ DICHIARAZIONE di TIPI LIBRERIA

DICHIARAZIONE di FUNZIONI LIBRERIA

INCLUDE di LIBRERIE ESTERNE (e.g. cmath)

```
void somma (double v1[], double v2[], double res[])  
{  
    for (int i = 0; i < DIM; i++)  
        res[i] = v1[i] + v2[i];  
}
```

...

```
double prodScale(double v1[], double v2[]) {  
    double accu = 0.;  
    for (int i = 0; i < DIM; i++) accu += v1[i] * v2[i];  
    return accu;  
}
```

```
double modulo(double v[]) {  
    return sqrt(prodScale(v, v));  
}
```

...

FILE OGGETTO : vettoriR2.o ← estensione  
nome libreria

⚠ Questo file è GENERATO dal COMPILATORE

g++ -c vettoriR2.c

se NON ci sono errori sintattici:

↓  
vettoriR2.o

L'opzione -c dice al compilatore di creare un file contenente la descrizione dei record di attivazione di tutte le funzioni DEFINITE nel file .c che passiamo.

Il prodotto è un file oggetto CHE NON è ESEGUIBILE



## USO DI LIBRERIE

ad es. `programma.c`  
Un programma (ovvero file con un `main`) che volesse usare la libreria DEVE:

- ① includere il file header delle librerie  
`#include "vettori2.h"` ! si trova nella stessa cartella del programma, altrimenti  
    / path relativo  
    \ " assoluto

- ② Essere compilato con

`g++ -c programma.c`  
↓  
`programma.o`

`programma.o`: file OGGETTO contenente la descrizione dei record di attivazione delle funzioni/procedure definite in `programma.c`

A questo punto abbiamo:

vettori R2.o

programmi.o : contengono tutti i record di attivazione  
di tutte le funzioni.

③ Collego tutto per creare un singolo file eseguibile  
che conterrà:

- descrizione record main
- descrizione dei record delle funzioni usate
- [ - informazioni accessorie ]

g++ programma.o vettori R2.o  
elenco file oggetto necessari

- o programma.o  
nome del file  
ESEGUIBILE prodotto  
della compilazione  
[linking dei .o]