

# Introduzione al Linguaggio C: esercitazioni

Federico Ficarelli

Cineca, 17-18 Maggio 2012

A decorative graphic element consisting of several horizontal lines of varying lengths and colors (teal, light blue, white) extending from the right side of the slide towards the center.

# Note utili

- Archivio degli esercizi:  
<http://corsi.cineca.it/courses/corsoC/esercizi.tar.bz2>
- Ottime fonti per dubbi *veloci*:
  - Wikipedia
  - GNU Manpages (Linux)
  - [www.cppreference.com](http://www.cppreference.com)
  - [stackoverflow.com](http://stackoverflow.com)
  - «*Google is your friend*»

# Esercizio 1

- E' fornito un piccolo programma che non compila (incompleto).
- Strutturato in fasi:

lavorare in modo incrementale, verificando che dopo ogni fase il programma compili e continui a funzionare correttamente!

- Il programma è (per ora) composto da un solo file: potete scrivere il vostro Makefile o compilare a mano.

Es.:

```
gcc -Wall -pedantic -Werror -O0 -g -o esercizio1.x esercizio1.c
```

# Esercizio 1: fase 1

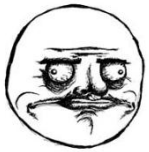
- Completare le parti mancanti contrassegnate nel sorgente con il tag #FASE1
- Richiede:
  - Preprocessore
  - Loop

# Esercizio 1: fase 2

- Completare le parti mancanti contrassegnate nel sorgente con il tag #FASE2
- Richiede:
  - Prendere confidenza con le funzioni
  - Eliminazione del codice ridondante

# Esercizio 1: fase 3

- Split del sorgente.
- Richiesto:
  - Ogni funzione deve risiedere nel **proprio file C**
  - Ad ogni file C deve corrispondere un **header** con **dichiarazione ed header-guard**.
  - Il file originale dovrà contenere solo la funzione main.
- Opzionale, ma gradito:
  - Scrivete il vostro build-system!
  - Gli header file dovranno stare in un'apposita directory «include».



# Esercizio 2

- Si parte dall'Esercizio 1 (completato).
- Anche questo va svolto in modo incrementale.

## Esercizio 2: fase 1

- Nell'esercizio precedente, i vettori erano automatici.
- Richiesto:
  - Scrivere una funzione di allocazione per vettori 1d
  - Scrivere una funzione di deallocazione
  - Rendere dinamici i vettori trattati dal programma (usando le funzioni appena scritte)



## Esercizio 2: fase 2

- Aggiungere una dimensione agli array
- Richiede:
  - Decidere il tipo dei puntatori (gli array stessi).
  - Modifica di tutte le funzioni esterne per gestire due dimensioni.
  - Modifica delle funzioni di allocazione e deallocazione.
  - Leggere dimensioni da riga di comando (vedere la funzione `parse_cmd` fornita insieme ai sorgenti)

## Esercizio 3: opzionale

- Scrivere una funzione che esegua la moltiplicazione di due matrici arbitrarie
- Richiede:
  - Il risultato deve essere scritto e restituito in una matrice allocata ad-hoc dalla funzione stessa.
  - La funzione deve essere «solida» (es.: segnalare errore se non riesce ad allocare o le dimensioni degli operandi non sono compatibili).