

# Esame linguaggio C

## Introduzione

Implementare i seguenti esercizi in linguaggio C.

Condividere il risultato sul proprio repository di github.

Creare una cartella **esame** e per ogni esercizio una sotto cartella con il nome dell'esercizio (Esercizio1, Esercizio2)

## Esercizi

### Esercizio 1 (5 punti)

Creare una funzione che accetti in input un array di interi e restituisca la somma dei valori contenuti nell'array partendo dall'indice 0 fino a che non viene trovato un valore negativo.

Esempio: [2,4,6,-2, 1] = 12

Stampare a console il risultato ottenuto a partire dal seguente array:

```
int array[] = {1,4,5,2,6,98,546,123,71,-3,44,5}
```

### Esercizio 2 (5 punti)

Implementare due funzioni: zip e somma.

La funzione zip accetta due array di interi a e b della stessa lunghezza e crea un terzo array che contiene i valori dei precedenti array intervallati tra di loro (partendo dal primo elemento dell'array a).

Esempio (in pseudocodice):

a = [1,7,9,4]

b = [4,5,6,9]

Risultato array:

risultato: [1,4,7,5,9,6,4,9]

La funzione somma, restituisce la somma degli elementi dell'array di interi passato come parametro di input.

Implementare la prima funzione (zip) su un file denominato: zip.c e dichiararla su un file zip.h

Implementare la seconda funzione (somma) su un file denominato: somma.c e dichiararla su un file somma.h

Creare infine un file main.c contenente la funzione main che stampa a console il risultato della somma dello zip di due array.

Testare la funzione creando un array contenente i primi 100 numeri interi pari [0,2,4,6,8,...] e un array contenente i primi 100 numeri interi dispari [1,3,5,7,9,11,13,...].  
Stampare a console il risultato ottenuto.

# Criteri di assegnazione voto

Il voto terrà conto dei seguenti parametri:

Sintassi (20%)

- utilizzo di nomi delle variabili e funzioni significative
- identazione secondo le convenzioni viste in classe
- uso di eventuali commenti adeguato

Semantica (45%)

- uso appropriato dei costrutti di C e in particolare di:
  - passaggio parametri funzioni,
  - puntatori, strutture
  - creazione memoria dinamica

Esecuzione e Test (35%)

- correttezza risultato