

# Parziale di Programmazione I - BioInformatica

1 febbraio 2019 (tempo disponibile: 2 ore)

## Esercizio 1 (9 punti)

Si scriva un programma `sum.c` che implementa una funzione `int sum(int arr[], int length)`. Tale funzione deve ricevere un array `arr` di interi, lungo `length`, e deve restituire la somma degli elementi di `arr` che abbiano l'elemento precedente dispari. Per esempio, se `arr` fosse `{2, 3, 4, 1, 5}`, la funzione dovrebbe restituire 9 (la somma di 4 e 5). Si scriva il file di header `sum.h` in cui si dichiara tale funzione.

## Esercizio 2 (11 punti)

Si scriva un programma `main_sum.c` che include la funzione dell'Esercizio 1 tramite il file di header `sum.h`. Il programma `main_sum.c` deve contenere una funzione iniziale `main` che esegue le seguenti operazioni:

1. legge da tastiera la lunghezza `length` di un array, richiedendola ad oltranza se fosse inserita negativa;
2. crea un array `elements` di `length` interi;
3. legge da tastiera gli elementi di tale array, uno alla volta;
4. chiama la funzione `sum` dell'Esercizio 1, passando `elements` e `length`;
5. stampa sul video il risultato di tale chiamata.

## Esercizio 3 (12 punti)

Si scriva un programma `cross.c` con una funzione iniziale `main` che esegue le seguenti operazioni:

1. legge da tastiera una dimensione intera `size`, dispari e positiva. Se non lo fosse, la richiede ad oltranza;
2. stampa su video una croce di asterischi, le cui aste hanno lunghezza `size`.

Per esempio, se l'utente inserisse 5 come `size`, il programma dovrebbe stampare:

```
*
*
*****
*
*
```