

# Parziale di Programmazione I - BioInformatica

5 febbraio 2020 (tempo disponibile: 2 ore)

## Esercizio 1 (13 punti)

Si scriva un programma `has_local_max.c` che implementa una funzione `int has_local_max(int arr[], int length, int how_many)`. Tale funzione deve ricevere un array `arr` di interi, lungo `length`, e deve determinare se tale array ha almeno `how_many` massimi locali. Un massimo locale è un elemento che è maggiore sia del precedente che del successivo (se esistono). Per esempio, se `arr` fosse  $\{12, -1, 10, 7, 5, 6\}$ , i suoi massimi locali sarebbero 12, 10 e 6. Si scriva il file di header `has_local_max.h` in cui si dichiara tale funzione.

## Esercizio 2 (10 punti)

Si scriva un programma `main_local_max.c` che include la funzione dell'Esercizio 1 tramite il file di header `has_local_max.h`. Il programma `main_local_max.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 `has_local_max` dell'Esercizio 1, per sapere se l'array `elements` contiene almeno 3 massimi locali;
5. sulla base del risultato di tale chiamata, stampa "Ci sono almeno tre massimi locali" oppure "Ci sono meno di tre massimi locali".

## Esercizio 3 (9 punti)

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

1. legge da tastiera una dimensione intera `size` positiva. Se non lo fosse, la richiede ad oltranza;
2. stampa su video il seguente disegno, le cui aste hanno lunghezza `size`.

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

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