

# Compito di Programmazione - Bioinformatica

5 luglio 2023 (tempo disponibile: 2 ore)

## Esercizio 1 (15 punti) (si consegnì bits.c)

Un numero naturale è detto avere ripetizione due se nella sua rappresentazione binaria ci sono almeno due bit ad 1 consecutivi. Per esempio, il numero 22 ha ripetizione due poiché la sua rappresentazione binaria 10110 ha almeno due bit ad 1 consecutivi. Similmente, il numero 92 ha ripetizione due poiché la sua rappresentazione binaria 1011100 ha almeno due bit ad 1 consecutivi. Invece il numero 18 non ha ripetizione due poiché la sua rappresentazione binaria 10010 non ha almeno due bit ad 1 consecutivi.

Si completi il seguente file `bits.c` (si noti che la funzione `ripetizione2` deve essere ricorsiva):

```
// AGGIUNGERE QUI GLI #include NECESSARI

// riempie arr con interi casuali, divisibili per 3, tra 0 e 65535 inclusi;
// arr ha lunghezza length
void init_random(int arr[], int length) {
    // COMPLETARE
}

// determina se n ha ripetizione due, cioè se la sua rappresentazione
// binaria contiene almeno due bit ad 1 consecutivi;
// si dia per scontato che n sia tra 0 e 65535 inclusi;
// questa funzione deve essere ricorsiva
int ripetizione2(int n) {
    return 0; // MODIFICARE E COMPLETARE, DEVE ESSERE RICORSIVA
}

// ordina gli elementi di arr, lungo length, in modo da spostare all'inizio
// i suoi elementi che hanno ripetizione due e alla fine i suoi
// elementi che non hanno ripetizione due
void sort_by_ripetizione2(int arr[], int length) {
    // COMPLETARE
}
```

**I file `bits.h` e `main.bits.c` sono già scritti e completi, non vanno modificati e non vanno consegnati. Se servisse, si possono aggiungere funzioni ausiliarie dentro `bits.c`.**

Se tutto è corretto, un esempio di esecuzione di `main.bits.c` potrebbe essere:

```
Inserisci la lunghezza dell'array, non negativa: 100
60417 (ripetizione2=1)
7842 (ripetizione2=1)
36831 (ripetizione2=1)
42237 (ripetizione2=1)
.....
```

```
30150 (ripetizione2=1)
47757 (ripetizione2=1)
42273 (ripetizione2=0)
4362 (ripetizione2=0)
38226 (ripetizione2=0)
36936 (ripetizione2=0)
```

## Esercizio 2 (16 punti)

(si consegna es2\_lista\_multipli3.c)

Si consideri il file allegato `es2_lista_multipli3.c`, riportato per comodità anche qui sotto. Si completi la funzione `crea_lista_multipli_3()` in modo che, dato un numero intero  $N$  maggiore di 100, generi una lista contenente tutti i numeri positivi multipli di 3 e minori di  $N$ . Si completi anche la funzione `calcola_media()` in modo che, data una lista di interi, calcoli la media aritmetica di tutti i valori della lista. Si usi il `main` già scritto per testare le due funzioni. **Si** possono aggiungere altre funzioni ma **non** si deve modificare il `main`.

```
#include <stdio.h>
#include <stdlib.h>

struct nodo *crea_lista_multipli_3(struct nodo *head, int valore);
float calcola_media(struct nodo *head);
struct nodo {
    int value;
    struct nodo *next;
};

// Program entry point
int main() {
    int n = 150;
    struct nodo* head = (struct nodo *)malloc(sizeof(struct nodo));
    head=crea_lista_multipli_3(head,n);
    printf("La media e' %f", calcola_media(head));

    return 0;
}

/**
 * Crea un lista con i multipli di 3 positivi e inferiori a n
 */
struct nodo *crea_lista_multipli_3(struct nodo *head, int n) {
    //da completare
}

/**
 * Restituisce la media degli elementi di una lista
 */
float calcola_media(struct nodo *head) {
    //da completare
}
```