



## ***Esercitazione di laboratorio n. 3***

### **Esercizio n.1: Elemento maggioritario**

*Competenze: ricorsione matematica (Ricorsione e problem-solving: 2.2)*

Sia dato un vettore `vet` di  $N$  naturali. Si definisce elemento maggioritario, se esiste, quel valore che ha numero di occorrenze  $> N/2$ .

Esempio: se  $N=7$  e `vet` contiene 3, 3, 9, 4, 3, 5, 3 l'elemento maggioritario è 3. Se  $N=8$  e `vet` contiene 0, 1, 0, 2, 3, 4, 0, 5 non esiste elemento maggioritario.

Si scriva una funzione `maggioritario` che, dati  $N$  e `vet`, visualizzi l'elemento maggioritario se esiste, -1 se non esiste. Il prototipo sia:

```
int majority( int *a, int N );
```

Vincoli: si risolva mediante ricorsione. Non è lecito utilizzare né algoritmi di ordinamento, né algoritmi di complessità  $O(N)$  che si possono trovare in rete.

### **Esercizio n.2: Anagrafica con liste**

*Competenze: creazione e gestione di liste concatenate (Puntatori e strutture dati dinamiche: 4.1)*

I dettagli di una anagrafica sono memorizzati in file di testo composti da un numero indefinito di righe nella seguente forma:

```
<codice> <nome> <cognome> <data_di_nascita> <via> <citta'> <cap>
```

Il campo `<data_di_nascita>` è nella forma gg/mm/aaaa, `<cap>` è un numero intero, mentre tutti i campi rimanenti sono stringhe senza spazi di massimo 50 caratteri. `<codice>` è nella forma AXXXX, dove X rappresenta una cifra nell'intervallo 0-9, ed è univoco nell'intera anagrafica. I dettagli dell'anagrafica vanno racchiusi in un opportuno tipo di dato `Item`.

L'anagrafica va memorizzata in una lista ordinata per anzianità (le persone più anziane **appaiono prima nella lista**).

Si scriva un programma in C che, una volta inizializzata una lista vuota, offra le seguenti funzionalità:

- acquisizione ed inserimento ordinato di un nuovo elemento in lista (da tastiera)
- acquisizione ed inserimento ordinato di nuovi elementi in lista (da file)
- ricerca, per codice, di un elemento
- cancellazione (con estrazione del dato) di un elemento dalla lista, previa ricerca per codice
- cancellazione (con estrazione del dato) di tutti gli elementi con date comprese tra 2 date lette da tastiera. Si consiglia, anziché di realizzare una funzione che cancelli dalla lista questi elementi, restituendoli memorizzati in una lista o in un vettore dinamico, di implementare una funzione che estragga e restituisca al programma chiamante il primo degli elementi appartenenti all'intervallo. Il programma chiamante itererà la chiamata di questa funzione, stampando il risultato, per tutti gli elementi dell'intervallo
- stampa della lista su file.

Per le funzioni di ricerca e cancellazione è richiesto che la funzione che opera sulle liste ritorni l'elemento trovato o cancellato al programma chiamante, che provvede alla stampa.

**Valutazione: tutti gli esercizi saranno oggetto di valutazione**

**Scadenza caricamento di quanto valutato: entro le 23:59 del 25/10/2025.**