

Compito di Programmazione - Bioinformatica

8 Luglio 2025 (tempo disponibile: 2 ore)

Esercizio 1 (31 punti)

(si consegnino i file `list.h`, e `list.c` opportunamente modificati)

Si vuole gestire un elenco di giocatori (*players*) per un torneo composto da un numero fissato `SCORE_MAX` di partite. Ciascun giocatore è rappresentato da una struttura contenente: un identificativo intero, una stringa `nickname` di lunghezza massima `CHAR_MAX` e l'elenco dei punteggi ottenuti al termine delle `SCORE_MAX` partite disputate (cioè un array di `float` di lunghezza `SCORE_MAX`). Il programma dovrà gestire i dati dei giocatori tramite una lista singolarmente concatenata non necessariamente ordinata.

La funzione `main` è già scritta e completa, non va modificata.

Completare invece i file `list.h` e `list.c` implementando le funzioni specificate. È consentito esclusivamente l'utilizzo delle funzioni appartenenti alle librerie già incluse nel codice fornito. Se ritenuto necessario, si possono aggiungere funzioni ausiliarie.

In particolare si completino le funzioni:

- `newPlayer()`: crea e inizializza un nuovo giocatore a partire da identificativo, `nickname` e punteggi.
- `insertPlayer()`: inserisce un giocatore **in coda** alla lista.
- `listLen()`: calcola la lunghezza della lista di giocatori fornita, in maniera **iterativa**.
- `getMean()`: calcola la media di un array di punteggi fornito, lungo `SCORE_MAX`;
- `printList()`: stampa l'elenco dei giocatori della lista fornita, nel formato: identificativo, `nickname`, array dei punteggi e punteggio medio.
- `insertInOrder()`: inserisce un nuovo giocatore nella lista fornita, in base all'ordinamento decrescente per media dei punteggi. In questo caso si assume che la lista fornita sia già decrescente per media dei punteggi.
- `createTopPlayersList()`: partendo dalla lista passata come argomento, crea una nuova lista contenente copie dei migliori giocatori, ordinata in maniera decrescente per media dei punteggi e contenente solo i giocatori con punteggio medio maggiore o uguale a un certo valore soglia `mediaTop`. **Questa funzione deve essere ricorsiva. Attenzione:** i nodi della nuova lista devono essere **copie** dei nodi originali, non riferimenti agli stessi elementi.
- `destroyList()`: dealloca la memoria occupata dall'intera lista fornita.

Se tutto è corretto, l'output del programma sarà il seguente:

Stampa lista inserita:

ID: 0, Nickname: FastLane, Best Scores: 7.00 8.50 9.00 8.50, Media: 8.25
ID: 1, Nickname: StackMaster, Best Scores: 6.00 7.00 7.50 7.00, Media: 6.88
ID: 2, Nickname: LoopHunter, Best Scores: 7.50 8.50 8.50 7.50, Media: 8.00
ID: 3, Nickname: NullPointer, Best Scores: 5.00 6.00 5.50 6.50, Media: 5.75
ID: 4, Nickname: ZeroLag, Best Scores: 4.00 5.00 6.00 5.50, Media: 5.12
ID: 5, Nickname: CodeSurfer, Best Scores: 9.00 8.50 9.50 8.00, Media: 8.75
ID: 6, Nickname: BugFinder, Best Scores: 3.00 6.50 7.00 6.00, Media: 5.62

Lunghezza della lista: 7

Lista dei top players con media superiore a 8.00, decrescente per media:

ID: 5, Nickname: CodeSurfer, Best Scores: 9.00 8.50 9.50 8.00, Media: 8.75
ID: 0, Nickname: FastLane, Best Scores: 7.00 8.50 9.00 8.50, Media: 8.25
ID: 2, Nickname: LoopHunter, Best Scores: 7.50 8.50 8.50 7.50, Media: 8.00

Deallocazione lista...

Deallocazione lista dei top players...