Informatica Generale 2a UD, A.A. 2009-2010 **27 Gennaio 2011**

Prova pratica di programmazione

Nome: _____

Cognome: _____

Matricola: _____

N.B.: Leggere attentamente tutto il testo dell'esercizio prima di svolgerlo.

Seconda unità - tempo stimato: 1h e 30m Valutazione: _____

Implementare una struttura dati dinamica di tipo "lista", chiamata `lista_t`. Ciascun elemento della lista contiene un dato di tipo `char *`. Creare due liste di tipo `lista_t` chiamate `cognomi` e `nomi`, per memorizzare separatemente i nominativi di una serie di persone letti da tastiera.

Implementare una funzione `int iniziali(lista_t* ini, lista_t* nom, lista_t* cog)` che estrapola dalle liste `nom` e `cog` le iniziali di nomi e cognomi, rovesciando l'ordine di inserimento. In particolare:

- Si supponga che ogni nominativo sia letto da tastiera nell'ordine `Cognome Nome(i)` ovvero che ogni persona abbia solo un cognome e almeno un nome.
- Le liste `cognomi` e `nomi` devono mantenere un ordinamento secondo inserimento e ammettono duplicati. Ad esempio, dati in input i nominativi `rossi mario, bianchi carlo paolo, verdi mario`, le liste conterranno rispettivamente `[rossi, bianchi, verdi]` e `[mario, carlo paolo, mario]`.
- La funzione `iniziali` ritorna il numero di nominativi da cui sono state estratte le iniziali.
- La lista `ini` deve mantenere un ordinamento decrescente secondo inserimento e contenere le iniziali (in maiuscolo) dei nomi e cognomi ricomposti, mettendo prima le iniziali del nome o dei nomi e poi l'iniziale del cognome. Ad esempio, dopo aver ricomposto le due liste citate sopra, la lista deve contenere gli elementi `[MV, CPB, MR]` e ritornare il valore `3`.
- La lista `ini` non ammette la presenza di duplicati. Ad esempio, se avessimo inserito un'ultimo nominativo `verdi marco`, ricomponendo le liste avremmo comunque ottenuto la lista di iniziali `[MV, CPB, MR]`, ma la funzione avrebbe ritornato `4`.

Implementare inoltre le funzioni:

- `void stampa(lista_t*)`, che stampa il contenuto della lista;
- `int dim(lista_t*)`, che restituisce il numero di elementi nella lista.
- `void libera(lista_t*)`, che libera la memoria occupata dagli elementi nella lista.

Implementare una funzione `main` che inizializza e popola le liste mediante `scanf`.

Dopodiche' chiama la funzione `iniziali` e stampa il contenuto delle tre liste e le rispettive dimensioni. Si stampi anche il valore ritornato dalla funzione `iniziali`.

Invocare poi la funzione `libera` sulle tre liste e riprovare ad invocare la funzione `iniziali`. A questo punto le tre liste dovrebbero essere vuote, quindi `iniziali` dovrebbe ritornare `0`, come anche la funzione `dim` richiamata sulle tre liste. Effettuare una stampa di controllo per verificare questi valori.

Suggerimenti

- Non implementare funzioni non necessarie o non esplicitamente richieste.
- E' possibile utilizzare la funzione `char toupper(char x)` (ctype.h);