Compito d'esame

19 Gennaio 2007

...alcune leggende metropolitane
dicono sia uno dei più tosti...

1

Avvertenze

- **Prima di cominciare:** si scarichi il file **StartKit2.1.zip** contenente il file di testo **utenti.txt** e **foto.txt**.
- **Avvertenze per la consegna:** nominare i file sorgenti come richiesto durante il testo del compito, apporre all'inizio di ogni file sorgente un commento contenente i propri dati (**cognome, nome, numero di matricola**) e il **numero** della prova d'esame. Al termine, **consegnare tutti i file sorgenti** ed i file contenuti nello StartKit.
- Rispettare **ALLA LETTERA** le specifiche, in particolare inserire le funzioni nei file specificati fra parentesi dopo il nome della funzione. Chi non rispetta le specifiche sarà opportunamente penalizzato. **NON SARANNO CORRETTI** gli elaborati che presenteranno un numero "non affrontabile" di errori di compilazione.

2

Introduzione

- Lo studio fotografico *FotoPerTutti* desidera mantenere un archivio delle proprie foto. L'applicazione mantiene, per ogni foto, le seguenti informazioni:
  - il nome della foto (unico all'interno del sistema) – al più 15 caratteri;
  - il nome utente del fotografo – al più 15 caratteri;
  - uno stato ('F' per pubblico, 'R' per privato) – un carattere;
  - un certo numero di parole chiave (fino ad un massimo di 10) per l'indicizzazione della foto – ogni parola chiave contiene al più 15 caratteri.
  - Per ogni fotografo, l'applicazione mantiene le informazioni relative a nome e password.
- Sull'archivio delle foto, ogni fotografo può effettuare delle interrogazioni (recuperare foto) su tutte le foto pubbliche e sulle proprie foto private. I dati (foto e utenti) sono memorizzati in file di testo (nello **StartKit**) di cui non si conoscono a priori le dimensioni. Ogni riga del file degli utenti è formato da due campi, nome utente e password entrambi al più di 15 caratteri e non contenenti spazi, separati dal carattere spazio. Ogni riga del file delle foto è formato al più da tredici campi separati dal carattere **';'** (tre per foto, fotografo e stato ed al più dieci per le parole chiave).

3

Esercizio 1 – Autenticazione autenticazione.c/.h

- Progettare una funzione di nome **autentica** (da invocare nel main come prima istruzione), che prenda in ingresso il nome del file degli utenti, che richieda all'utente nome e password, che verifichi direttamente sul file (il cui nome è passato come parametro) se la coppia nome e password è corretta e che in caso di verifica positiva restituisca il nome dell'utente autenticato, in caso di verifica negativa restituisca NULL.

4

Esercizio 1 – Verifica di funzionamento (non nel compito)

- Come verificare il funzionamento di una funzione?

- Generalmente:
  - Verificare casi “normali”
  - Verificare casi limite
    - Nello specifico: primo e ultimo utente nel file, file vuoto (autenticazione fallisce sempre)...

5

Esercizio 2 – Dati e lettura file foto.c/.h

- Progettare la struttura dati di nome **foto** atta ad ospitare le foto (si suggerisce di utilizzare una ulteriore struttura contenente un array di parole e un intero contenente la dimensione logica dell’array, atta a contenere le parole chiave) ed una funzione **leggiFoto (foto.c/foto.h)** che prenda in ingresso il nome di un file e che restituisca una lista contenente tutte le foto lette. Si implementi manualmente la lista usando direttamente i puntatori senza utilizzare l’ADT lista; la funzione **leggiFoto** deve utilizzare una funzione **aggiungiFoto (foto.c/foto.h)** – anch’essa da progettare – che prenda in ingresso una foto ed una lista di foto e che aggiunga la foto alla lista. La lettura delle foto deve essere effettuata solo in caso in cui il login dell’utente sia avvenuto con successo. Per la lettura dei campi si suggerisce di utilizzare la funziona **readField** modificata contenuta nello StartKit. Tale funzione, a differenza di quella studiata, restituisce il separatore effettivamente incontrato in modo da distinguere agevolmente fra il separatore specificato, il carattere di fine linea o il carattere di fine file.

6

Esercizio 3 – Menù
menu.c/.h

- Utilizzare nel **main** la funzione **mostraMenu (menu.c/menu.h)** che richiede all'utente quale azione voglia compiere. La funzione **mostraMenu** restituisce un valore intero che indica l'opzione scelta. In base a tale valore numerico, invocare opportunamente le funzioni da inserire in **menu.c/menu.h** che sono progettate in seguito. Il menù deve essere mostrato dopo la lettura delle foto quindi anch'esso solo in caso di login avvenuto con successo. Continuare a mostrare il menù fintanto che l'utente non richieda di uscire.

7

Esercizio 4 – Ordinamento foto
foto.c/.h + menu.c/.h

- Progettare una funzione **menuOrdinaFoto (menu.c/menu.h)** che prenda in ingresso la lista delle foto e l'utente corrente (quello loggato), crei una nuova lista con le sole foto dell'utente (pubbliche o private) ordinate secondo il nome della foto in ordine lessiografico ascendente e stampi la lista. L'ordinamento deve essere effettuato dalla funzione **ordinaFoto (foto.c/foto.h)** che deve prendere in ingresso la lista delle foto e il nome dell'utente e deve restituire la lista ordinata. Si effettui l'ordinamento tramite inserimento ordinato nella nuova lista; l'inserimento ordinato deve essere fattorizzato nella funzione **insOrdFoto (foto.c/foto.h)** che deve prendere in ingresso una lista di foto (in cui inserire) ed una **foto** e deve restituire una lista in cui la **foto** sia stata inserita in modo ordinato. Al fine di facilitare eventuali modifiche del codice, si racchiuda il criterio di ordinamento in una funzione **compareFoto (foto.c/foto.h)** che prenda in ingresso due **foto** e restituisca un intero positivo nel caso in cui la prima foto sia maggiore della seconda, nullo in caso di uguaglianza, negativo altrimenti; il confronto deve essere effettuato in base al criterio specificato sopra. La stampa della lista risultante dall'ordinamento deve essere effettuata tramite la funzione **stampaFoto (foto.c/foto.h)** che stampa sullo schermo tutte le foto incluse nella lista che viene passata come parametro.

8

Esercizio 5 – Ricerca foto
foto.c/.h + menu.c/.h

- Progettare una funzione **menuRicercaFoto (menu.c/menu.h)** che prenda in ingresso la lista delle foto e l'utente corrente (quello loggato), richieda l'inserimento di una parola chiave per la ricerca, utilizzi la funzione **ricercaFoto (foto.c/foto.h)** per la ricerca delle foto e stampi a video il risultato della ricerca. La funzione **ricercaFoto** deve prendere in ingresso la lista di foto, il nome dell'utente corrente e la parola chiave e deve restituire una lista di foto scegliendo fra le foto pubbliche di tutti gli utenti e quelle private dell'utente corrente e che contengano la parola chiave richiesta. Per la stampa, utilizzare la funzione **stampaFoto** (**foto.c/foto.h)** progettata nell'esercizio precedente.

9

The end

- Il compito è finito...
- ...era veramente difficile?
- ...oppure sarebbe bastato leggere bene il testo?

10

Extra 1 – Inserimento manuale

- Estendere l'applicazione in modo da rendere possibile l'inserimento manuale di nuove foto.
- Aggiungere:
  - Una voce di menù e relativa funzione di gestione
  - Una funzione di lettura da console che sia *user friend*:
    - Guida nell'inserimento dei parametri
    - Guida nell'inserimento delle parole chiave – le parole chiave (max. 10) devono essere lette finché l'utente non inserisce la stringa vuota

11

Extra 2 – Vincolo di unicità

- Fare in modo che l'applicazione verifichi il vincolo di unicità sui nomi delle foto (come previsto nell'introduzione) – tale vincolo deve essere rispettato sia per foto inserite in modo manuale, sia per foto lette da file.
  - Dove far rispettare il vincolo?
  - Cosa fare in caso di violazione?
  - Come vengono aggiornate le funzioni di lettura da file e di lettura da console?

12