

Fondamenti di Informatica e Laboratorio T-AB  
Prova Pratica – 12 Gennaio 2015  
Compito

**Prima di cominciare:** si scarichi il file **StartKit.zip** contenente i file di esempio.

**Avvertenze per la consegna:** nominare i file sorgenti come richiesto nel 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 sorgente** ed i file contenuti nello StartKit.

**Rispettare 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 ragionevole" di errori di compilazione.

Il negozio *GoodReading* dispone di un avanzato sistema informatizzato che aiuta il personale nella gestione dell'inventario e nella gestione delle opinioni dei clienti. Il sistema memorizza i prodotti in un opportuno file *prodotti.txt* (presente nello start kit). Ogni riga di tale file corrisponde ad un prodotto ed è formalizzata nel seguente modo:

**codiceProdotto autore nomeProdotto quantProdotto annoDistribuzione**

dove il codiceProdotto è una stringa di 5 caratteri, il nomeProdotto e autore sono stringhe (senza spazi) di al massimo 30 caratteri, quantprodotto e annoDistribuzione sono valori numerici interi.

### *Esercizio 1 - Lettura dei dati (libri.h/libri.c)*

Si definisca un'opportuna struttura dati, al fine di rappresentare i dati registrati nel file "**prodotti.txt**", come specificato sopra, e si realizzi una funzione:

**Prodotto\* readprodotti(char\* fileName, int \*dim);**

che, ricevuto in ingresso il nome di un file contenente i dati dei prodotti, legga i dati relativi, allochi la quantità di memoria necessaria a contenerli e ve li memorizzi. Tale funzione deve restituire un riferimento all'area di memoria allocata (array) e la sua dimensione logica (numero di prodotti letti) in **dim**. Si noti che non è noto a priori quanti prodotti siano presenti nel file: sarà quindi necessario determinare quanti ve ne siano, e poi allocare la memoria necessaria dinamicamente.

Si realizzi in seguito la funzione

**void stampaProdotto(Prodotto prod);**

che, ricevuto in ingresso un **Prodotto**, ne stampi a video il contenuto.

Inoltre, il sistema tiene memorizzato in un file di testo "**opinioni.txt**" l'elenco delle opinioni che i clienti del negozio hanno espresso. Ogni riga del file (vedi start kit) identifica un'opinione ed è composta nel seguente modo:

**codiceProdotto votoProdotto codiceCliente anno mese giorno**

dove il votoProdotto è un valore numerico intero, il codicecliente è una stringa (senza spazi) di al massimo 5 caratteri ed anno, mese e giorno sono valori interi.

Si definisca un'opportuna struttura dati e si realizzi poi una funzione

**Opinione \* leggiOpinioni(char\* fileName, int \*dim);**

che, ricevuto in ingresso il nome di un file contenente i dati, legga i dati, allochi la quantità di memoria necessaria a contenerli e ve li memorizzi. Tale funzione deve restituire un riferimento all'area di memoria allocata (array) e la sua dimensione logica (numero di opinioni lette) in **dim**.

Si realizzino nel main le opportune istruzioni per verificare il corretto funzionamento delle funzioni realizzate.

Fondamenti di Informatica e Laboratorio T-AB  
Prova Pratica – 12 Gennaio 2015  
Compito

*Esercizio 2 – controllo opinione (function.h/function.c)*

Il sistema informatizzato prevede che ogni cliente possa esprimere una sola opinione per ogni prodotto, per cui, in caso di multiple votazioni per lo stesso prodotto, il sistema dovrà tenere traccia dell'opinione corretta. In particolare, tale opinione è quella avente la data più recente.

Il candidato realizzi una funzione

**int controlloOpinione(Opinione\* list, int dim, Opinione op);**

che prenda in ingresso un array di opinioni **list** con la relativa dimensione **dim** ed un'opinione **op**. Tale funzione restituirà un valore da intendere come "vero" se e solo se l'opinione **op** è corretta. Pertanto la funzione dovrà verificare che, nel caso in cui il cliente autore dell'opinione **op** abbia votato più volte lo stesso prodotto, la data di votazione di **op**, passata all'interno della struttura **op** come parametro di input, sia la più recente fra tutte quelle relative allo stesso cliente e prodotto all'interno di **list** (per semplicità, si supponga NON ci siano conflitti sulle date).

Il candidato scriva le opportune istruzioni nel main per testare questa funzione.

*Esercizio 3 – conta dei libri (function.h/function.c)*

Il candidato realizzi una funzione

**int contaLibri(char\* autore, Prodotto\* p, int dim);**

che prenda in ingresso una stringa **autore**, l'insieme dei prodotti e la relativa dimensione **dim**. La funzione deve restituire la somma delle quantità dei prodotti dello stesso **autore**.

Il candidato scriva le opportune istruzioni nel main per testare questa funzione.

*Esercizio 4 – interrogazioni del database (function.h/function.c)*

Si realizzi inoltre una funzione

**Opinione \*opinioniAutore(Opinione\* o, int dimOp, Prodotto\* p, int dimProd, int \*dimOpAutore);**

che preso in ingresso un array di **Opinione** e di **Prodotto** di dimensione rispettivamente **dimOp** e **dimProd**, restituisca un'array (creato all'interno della funzione stessa) di **Opinione** contenente tutte e sole le opinioni espresse sui prodotti relativi ad un autore. Il nome dell'autore non è passato come parametro di input, quindi la funzione dovrà, al suo interno, chiedere all'utente il nome dell'autore di cui si vogliono conoscere le opinioni. La funzione restituisca infine in **dimOpAutore** il numero delle opinioni inserite nell'array restituito.

Il candidato scriva le opportune istruzioni nel main per testare la funzione e stampi a video l'array ottenuto.

*Esercizio 5 – inserimento di un prodotto (libri.h/libri.c)*

Il candidato realizzi una funzione

**void inserisciProdotto(char\* fileName);**

che aggiunga al file (il cui nome è contenuto nella stringa **fileName**) il prodotto correttamente formattato. I dati necessari per l'inserimento del prodotto devono essere chiesti all'utente. La funzione deve, inoltre, garantire che non vengano inseriti due prodotti aventi lo stesso codice.

**Suggerimento:** prima di aggiungere il prodotto, utilizzare le funzioni già definite per andare a leggere dal file l'elenco dei prodotti presenti per il confronto.

Il candidato scriva le opportune istruzioni nel main per testare questa funzione e si ricordi che per aggiungere un elemento ad un file esiste un modo apposito di apertura del file stesso.