

Laboratorio di Informatica L-A

Prova d'Esame 7 – 7 Settembre 2007

Prima di cominciare: si scarichi il file **StartKit7.zip** contenente i file necessari.

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 sorgenti** 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 affrontabile" di errori di compilazione.

Consiglio: per verificare l'assenza di *warnings*, effettuare di tanto in tanto un *Rebuild All*.

Una catena americana di pizzerie permette la prenotazione di pizze tramite il proprio sito web. Si deve realizzare un programma che, forniti un insieme di ordini di pizze ed il listino dei prezzi dei singoli ingredienti, calcoli la spesa da fatturare ad ogni cliente. Le informazioni sono contenute in due differenti file, denominati rispettivamente **ingredienti.txt** e **ordini.bin**.

Il file di testo **ingredienti.txt** memorizza, in ogni riga, il **nome** dell'ingrediente (al più 24 caratteri, senza spazi), ed il **prezzo** di tale ingrediente (un float).

Il file binario **ordini.bin** memorizza invece gli ordini: all'inizio è registrato il numero di ordini presenti in tutto il file (un intero). Di seguito, sono registrate tutte le pizze ordinate: esattamente in questo ordine, per ogni pizza, sono salvati il **nome** del cliente che ha ordinato la pizza (al più 64 caratteri), il **numero** di ingredienti presenti su quella pizza (al più 10), e un **array** di 10 elementi (non tutti usati) di ingredienti (composti dalla coppia nome ingrediente/prezzo). Ovviamente il file può contenere più pizze ordinate dalla stessa persona, e le liste degli ingredienti ivi registrati, anche se composte dalla coppia nome ingrediente/prezzo, in realtà riportano correttamente solo il nome dell'ingrediente: infatti il prezzo è soggetto a frequenti modifiche (fa testo in tal senso il solo listino prezzi ufficiale dato nel file precedente).

Esercizio 1 – Lettura/Scrittura degli ingredienti (listino.h/listino.c)

Il candidato realizzi un modulo per la gestione del listino degli ingredienti. In particolare, definisca una opportuna struttura dati **Ingrediente** (listino.h), per tenere traccia del **nome** di un ingrediente (al più 24 caratteri senza spazi) e del suo **prezzo** (un float). Il candidato poi definisca:

- Un insieme di primitive opportune per poter usare/gestire liste di strutture dati **Ingrediente** (si faccia riferimento, a tal scopo, alle primitive sulle liste viste a lezione) (listino.h/listino.c)
- Una funzione **leggiIngredienti(...)** che, ricevuto in ingresso il nome di un file, provveda ad aprire il file, e restituisca una lista di strutture dati **Ingrediente** lette da tale file. (si veda come esempio il file **ingredienti.txt** fornito nello StartKit).
- Una funzione **scriviIngredienti(...)** che, ricevuto come parametri d'ingresso il nome di un file di testo ed una lista di strutture dati **Ingrediente**, scriva le informazioni relative agli ingredienti sul file indicato. In particolare su ogni riga si scriva il nome dell'ingrediente e, separato da uno spazio, il suo prezzo. A titolo esemplificativo, il file ottenuto dall'invocazione dovrebbe essere del tutto simile al file **ingredienti.txt** fornito nello StartKit.

Al fine di verificare la corretta implementazione dell'esercizio 1, il candidato provveda ad utilizzare, nella funzione **main** (file main.c) le funzioni **leggiIngredienti** e **scriviIngredienti**, utilizzando come input il file **ingredienti.txt**, e come output un file di testo qualunque: il candidato avrà cura di controllare poi che il contenuto dei due file sia sufficientemente simile, e che non vi siano stati introdotti errori.

Esercizio 2 – Modifica dei prezzi (listino.h/listino.c)

Il candidato estenda il modulo definito al punto precedente realizzando:

- Una funzione **trovaPrezzo(...)** che, ricevuti in ingresso il nome di un ingrediente, ed una lista di strutture dati **Ingrediente**, restituisca il prezzo di tale ingrediente.

Laboratorio di Informatica L-A

Prova d'Esame 7 – 7 Settembre 2007

- b) Una funzione **aggiornaPrezzo(...)** che, ricevuti in ingresso la lista di strutture dati **Ingrediente**, il nome di un ingrediente ed un nuovo prezzo, restituisca la lista di ingredienti dove il prezzo dell'ingrediente specificato è stato aggiornato al nuovo valore. A tal scopo, il candidato consideri di accedere alla lista usando la notazione a puntatori.

Al fine di verificare la corretta implementazione dell'esercizio 2, il candidato provveda a modificare la lista di ingredienti letti nell'esercizio precedente e controlli, salvando la lista aggiornata su un file, che tale modifica sia avvenuta con successo (funzione **main**, file **main.c**).

Esercizio 3 – Gestione delle pizze (pizza.h/pizza.c)

Il candidato realizzi un modulo per la gestione di una singola pizza. In particolare, si definisca una struttura dati **Pizza** (file **pizza.h**) contenente le seguenti informazioni (esattamente in questo ordine): il **nome** del cliente che ha ordinato la pizza (al più 64 caratteri), il **numero** di ingredienti (un intero), ed un **array** di strutture dati **Ingrediente**, di dimensione statica di 10 elementi. Tipicamente una pizza conterrà meno di 10 ingredienti: il numero di tali ingredienti è indicato nell'apposito campo. Il candidato poi implementi:

- a) Una funzione **leggiPizza(...)** che, ricevuto in ingresso un puntatore a file binario (già opportunamente aperto), legga da tale file una struttura dati di tipo **Pizza** e la restituisca come risultato.
- b) Una funzione **calcolaPrezzo(...)** che, ricevuti in ingresso una struttura dati di tipo **Pizza**, ed una lista di strutture dati **Ingrediente**, calcoli il costo di tale pizza e lo restituisca come risultato della funzione.

Esercizio 4 – Gestione degli ordini (ordini.h/ordini.c)

Il candidato realizzi un modulo di gestione degli ordini, che sono registrati su un file binario (a titolo di esempio, si consideri il file **ordini.bin** nello StartKit). Su tale file viene salvato innanzitutto il numero di pizze ordinate (un intero), e poi a seguire delle strutture dati di tipo **Pizza**. Uno stesso cliente può ordinare più pizze, che possono essere registrate nel file anche in ordine non consecutivo; il file può contenere gli ordini relativi a diversi clienti. Il candidato implementi:

- a) Una funzione **leggiOrdini(...)** che, ricevuto in ingresso il nome di un file di ordini, provveda ad aprire tale file, a leggere il numero di pizze ivi registrate, ad allocare dinamicamente memoria sufficiente, e a leggere le strutture dati di tipo **Pizza** presenti in tale file. La funzione dovrà restituire un puntatore all'area di memoria allocata e, tramite un parametro passato per riferimento, il numero di pizze lette.
- b) Una funzione **sort(...)** che, utilizzando un algoritmo di ordinamento a scelta del candidato (tra quelli visti a lezione), ordini un array di strutture dati di tipo **Pizza** in base al nome del cliente che ha ordinato tale pizza.

Esercizio 5 – Calcolo della spesa relativa a ciascun cliente (main.c)

Il candidato, utilizzando le funzioni definite negli esercizi precedenti, provveda a leggere il listino degli ingredienti dal file **ingredienti.txt**, aggiorni il prezzo dell'ingrediente "**salame_piccante**" a 2.00 euro, e salvi (sovrascrivendo il file originale) la nuova lista di prezzi nello stesso file.

Quindi il candidato provveda a leggere dal file **ordini.bin** un array di pizze richieste da diversi clienti, ordini tale array in base al nome del cliente, e calcoli per ogni cliente la spesa totale (data dalla somma del costo di ogni pizza richiesta dallo stesso cliente). *Suggerimento: si sfrutti il fatto che l'array è ordinato proprio in base al nome del cliente, e che quindi pizze richieste dalla stessa persona risulteranno essere adiacenti*).