

Laboratorio di Informatica L-A

Prova d'Esame 1 – 20 Dicembre 2007

Prima di cominciare: si scarichi il file **StartKit1.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 famosa catena di librerie sta acquistando sul territorio piccole librerie ed intende fondere insieme i cataloghi. Poiché prima della fusione effettiva è necessario compiere alcune analisi di relazione fra i cataloghi, si deve realizzare un programma che, forniti i cataloghi di due librerie, consenta di stabilire quali siano i libri presenti in un catalogo e non nell'altro, i libri presenti in entrambi i cataloghi e consenta poi di unire i due cataloghi in un catalogo unico. Le informazioni relative ad un catalogo sono contenute in un file di testo (**biblio1.txt/biblio2.txt** nello start kit). Una riga di tale file di testo rappresenta un libro ed ha il seguente formato:

- Titolo – al più 50 caratteri + separatore ','
- Autore – al più 40 caratteri + separatore ','
- Genere – al più 30 caratteri + separatore ','
- Copie disponibili – intero

Poiché il committente ha intenzione di acquistare i codici sorgenti prodotti, ha dato incarico ad un ingegnere informatico della progettazione dell'applicazione, il quale, per avere la massima manutenibilità, ha imposto la scrittura di un ADT denominato **Set** che modelli un insieme e che possa ospitare degli **Element** da definire (in modo simile alle liste). Nel caso specifico gli **Element** dovranno essere libri.

Suggerimento: definire già da ora il tipo booleano in modo opportuno.

Esercizio 1 – Definizione dell'elemento dell'insieme (element.h/element.c)

Definire una struttura atta a contenere i dati relativi ad un libro di una libreria (v. sopra) e definire, tramite questa, il tipo di nome **Element**. Su tale tipo, definire le operazioni che seguono:

- Confronto: la funzione **compareElements** deve prendere in ingresso due **Element** e restituire un intero con la stessa semantica della funzione **strcmp**. Il confronto deve consentire di ordinare gli elementi prima per titolo alfabeticamente crescente poi, nel caso in cui i titoli siano uguali, per autore alfabeticamente crescente.
- Salvataggio di un elemento su file di testo: la funzione **saveElementToTxt** deve prendere in ingresso un puntatore ad file (aperto su un file di testo) e un elemento e deve salvare l'elemento sul file con la stessa sintassi di riga dei file contenuti nello start kit.
- Caricamento di un elemento da file di testo: la funzione **readElementFromTxt** deve prendere in ingresso un file (di testo) e un elemento (per riferimento) e caricare una riga del file di testo nell'elemento ricevuto. Tale funzione deve restituire un valore booleano che indica se la lettura è stata effettuata con successo o meno.
- Stampa a console di un elemento: la funzione **printElement** deve prendere in ingresso un elemento e stamparne a console le informazioni relative.

Scrivere nel **main** il codice di verifica del funzionamento del codice implementato. Tale codice deve leggere il primo libro (**Element**) di uno dei file di testo forniti nello start kit, scrivere il libro letto in un nuovo file di testo e stamparlo su console. Verificare anche il funzionamento della **compareElements** creando un nuovo **Element** e confrontandolo con quello caricato. Al termine del test commentare il codice senza eliminarlo.

Laboratorio di Informatica L-A

Prova d'Esame 1 – 20 Dicembre 2007

Esercizio 2 – operazioni di base sul Set (set.h/set.c)

Definire un ADT insieme di **Element** denominato **Set** implementato come una lista. Sul tipo insieme devono essere disponibili le operazioni “elementari” che seguono:

- Verifica di presenza di un elemento in un insieme: la funzione **setContainsElement** deve prendere in ingresso un **Set** e un **Element** e deve restituire un valore booleano che indichi se l'elemento specificato sia o meno presente nell'insieme specificato. Per i confronti fra elementi usare la funzione di confronto di cui all'esercizio 1.
- Aggiunta di un elemento ad un insieme: la funzione **addToSet** deve prendere in ingresso un **Set** ed un **Element** e deve aggiungere l'elemento all'insieme solamente nel caso in cui l'elemento non sia già contenuto nell'insieme stesso. La funzione deve restituire il **Set** elaborato.
- Conteggio degli elementi di un insieme: la funzione **countElementsInSet** deve prendere in ingresso un **Set** e restituire un intero che rappresenta il numero di elementi contenuti nel **Set** stesso.
- Stampa a console degli elementi di un insieme: la funzione **printSet** deve prendere in ingresso un **Set** e, tramite la funzione **printElement**, deve stampare a console tutti gli elementi contenuti nel **Set** stesso.

Nel main, si verifichi il funzionamento del codice scritto prima creando un nuovo libro (**Element**) e tentando di aggiungerlo due volte nello stesso **Set** (inizialmente vuoto), poi si effettui il conteggio degli elementi del **Set** verificando che il valore restituito sia 1, infine si stampi a console il contenuto del **Set**. Al termine del test commentare il codice senza eliminarlo.

Esercizio 3 – Lettura/scrittura Set da file di testo (set.h/set.c)

Il salvataggio di un **Set** su file di testo deve essere implementato nella funzione **saveSetToTxt**. Tale funzione deve prendere in ingresso un puntatore a file e un **Set** e deve salvare sul file tutti gli elementi (libri) contenuti nel **Set**. Per effettuare il salvataggio del singolo elemento, utilizzare la funzione **saveElementToTxt** precedentemente implementata. Il caricamento di un set da file di testo deve essere implementato nella funzione **readSetFromTxt**. Tale funzione deve prendere in ingresso un file ed un **Set** per riferimento e deve caricare nel **Set** tutti gli elementi (libri) presenti nel file di testo. Per effettuare il caricamento del singolo elemento, utilizzare la funzione **readElementFromTxt** precedentemente implementata.

Nel main, si verifichi il funzionamento del codice scritto caricando i file contenuti nello start kit, stampandoli a console (**printSet**) e scrivendoli su file diversi. Al termine del test commentare il codice senza eliminarlo.

Esercizio 4 – Differenza, unione, intersezione (set.h/set.c – main.c)

Definire le funzioni **setDifference**, **setUnion** e **setIntersection** in modo che prendano in ingresso due **Set** e restituiscano un nuovo Set come risultato dell'elaborazione. Dei due insiemi ricevuti in ingresso, la funzione **setDifference** deve calcolare e restituire l'insieme differenza (differenza = tutti e soli gli elementi del primo insieme che non sono contenuti nel secondo), la funzione **setUnion** deve calcolare e restituire l'insieme unione, la funzione **setIntersection** deve calcolare e restituire l'insieme intersezione. Nel **main**, scrivere il codice necessario per caricare i dati contenuti nello start kit (due insiemi), calcolare gli insiemi differenza, intersezione e unione, stamparli a console e stampare il numero di elementi contenuti in tali insiemi.