

Fondamenti di Informatica T-1, 2008/2009 – Modulo 2

Prova d'Esame 5 – 19 Giugno 2009 – tempo a disposizione 180 min

Prima di cominciare: si scarichi dal sito <http://esamix.labx> il file **StartKit.zip** contenente i file necessari (*solution* di VS2005 e progetto compresi).

Avvertenze per la consegna: 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** e 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.

Nota 1: NON SARANNO CORRETTI gli elaborati che presenteranno un numero "non affrontabile" di errori di compilazione.

Nota 2: il main non è opzionale; i test richiesti vanno implementati.

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

Un rigattiere mantiene in due file di testo tutti gli oggetti che compra e tutti gli oggetti che vende, rispettivamente `oggettiComprati.txt` e `oggettiVenduti.txt`. La prima riga di ciascun file contiene unicamente un intero che rappresenta la dimensione massima delle parole contenute in quel particolare file (utilizzare tale informazione per allocare la memoria in maniera opportuna). Ciascun file è composto di più righe e ciascuna riga (ad eccezione della prima) di più oggetti (sequenze di caratteri), separati tra loro da spazi. Una stringa che rappresenta un oggetto si dice ben formata quando non contiene spazi (usati solamente per la separazione fra nomi di oggetti diversi).

Esercizio 1 – Censimento oggetti venduti

Si realizzi una funzione che, data una stringa ben formata rappresentante un oggetto e un riferimento al file `oggettiVenduti.txt`, restituisca quante volte tale oggetto è stato venduto.

Esercizio 2 – Lista di oggetti venduti

Sfruttando quanto realizzato al punto precedente, si realizzi una funzione che restituisca la lista degli oggetti comprati che sono stati venduti almeno N volte, specificando per ogni oggetto il numero di volte che è stato venduto. A tal fine si definisca un'opportuna struttura dati che sia in grado di contenere sia il nome dell'oggetto, sia il numero di volte che tale oggetto è stato venduto.

Tale lista può contenere lo stesso oggetto ripetuto più volte. Ad esempio, fissando $N=2$ e se l'oggetto "boomerang" è stato comprato 4 volte e venduto 3, la lista restituita potrà contenere 4 volte l'oggetto "boomerang", ovvero tante quante l'oggetto comprato compare nel file degli oggetti comprati.

(Suggerimento: è possibile conoscere a priori la dimensione massima in caratteri del nome degli oggetti?)

Esercizio 3 – Ordinamento

Si invochi la funzione precedente allo scopo di ottenere tutti gli oggetti venduti almeno 3 volte. La lista ottenuta deve poi essere ordinata tramite ordinamento Bubble Sort (non ottimizzato e senza l'uso delle primitive). L'algoritmo deve ordinare la lista degli oggetti per numero di vendite crescenti, quindi dall'oggetto meno venduto a quello più venduto; se due oggetti sono stati venduti lo stesso numero di volte, allora si deve seguire l'ordine lessicografico sul nome del prodotto. La lista risultante DEVE contenere ciascun oggetto al più una volta.

Fondamenti di Informatica T-1, 2008/2009 – Modulo 2
Prova d'Esame 5 – 19 Giugno 2009 – tempo a disposizione 180 min

Esercizio 4 – Eliminazione efficiente dalla lista di oggetti venduti

Si eliminino dalla lista ottenuta al punto precedente tutti gli oggetti venduti un numero pari di volte. È importante che l'algoritmo realizzato sia efficiente e che liberi in modo definitivo la memoria relativa agli oggetti eliminati.

(Suggerimento: è possibile liberare memoria utilizzando unicamente le primitive?)

Esercizio 5 – Rovesciamento ricorsivo del file

Si realizzi un nuovo file di testo `rovesciato.txt` che contenga tutte le righe del file `oggettiComprati.txt` in ordine inverso, ad eccezione della riga con il numero intero che non deve comparire nel nuovo file. Si tenga conto del fatto che, per ipotesi, ciascuna riga è lunga al più 120 caratteri.

Ogni funzione realizzata deve essere opportunamente testata invocandola correttamente dalla funzione `main`.

Attenzione: la soluzione deve essere articolata su più file. La scelta di quanti e quali file utilizzare è a carico del candidato.