

Laboratorio di Informatica L-A

Prova d'Esame 3 – 9 Aprile 2008

Prima di cominciare: si scarichi il file **StartKit3.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*.

Si desidera realizzare un'applicazione che consenta di sottoporre esami a quiz (risposta multipla: una sola risposta corretta) agli studenti della facoltà. Ogni quiz è composto da una domanda e da un certo numero di risposte. Ogni risposta è composta di un codice di risposta (il codice che deve essere inserito dall'utente come codice di scelta), un punteggio e da un testo. L'applicazione deve leggere i quiz da un file di testo, deve sottoporre i quiz al candidato (un quiz alla volta) e deve calcolare il punteggio finale sommando i punteggi delle singole risposte date.

Un quiz deve essere modellato come segue:

- un testo della domanda: una stringa di al più 511 caratteri;
- una collezione di risposte (un array);
- una dimensione della collezione di risposte.

Una risposta deve essere modellata come segue:

- un codice: un carattere;
- un punteggio: un valore reale;
- un testo: una stringa di al più 511 caratteri.

Importante: la deallocazione della memoria è facoltativa – il non implementarla non darà luogo a decurtazione di punti mentre una corretta implementazione darà luogo ad un bonus.

Esercizio 1 – Definizione della struttura risposta e della collezione di risposte (answer.h/answer.c)

Definire una struttura atta a contenere i dati relativi ad una risposta (vedere sopra) e definire, tramite questa, il tipo di nome **Answer**. Definire una funzione di nome **createAnswerArray** che crei un array di **Answer** in memoria dinamica la cui dimensione è data da un parametro intero preso in ingresso dalla funzione stessa (il numero di risposte non è predeterminato). Definire una funzione di nome **readAnswerArrayFromTxt** che prenda in ingresso un puntatore a file e il numero di risposte da leggere (un intero) e che restituisca un array (allocato tramite **createAnswerArray**) contenente le risposte lette (verificare il formato delle risposte nel file **AnswerTest.txt**). Definire una funzione di nome **renderAnswer** che prenda in ingresso una risposta e ne stampi a *console* il codice ed il testo (non il punteggio).

Facoltativo: definire una funzione **destroyAnswerArray** che consenta di deallocare un array allocato tramite la **createAnswerArray**.

Nel **main**, si verifichi il funzionamento delle funzioni scritte leggendo il file **AnswerTest.txt** contenuto nello start kit e stampando a *console* (si usi **renderAnswer**) tutte le risposte lette – poiché **readAnswerArrayFromTxt** prende in ingresso il numero di risposte da leggere, si provveda a specificare il numero esatto di risposte contenute nel file. Al termine del test commentare il codice senza eliminarlo.

Esercizio 2 – Quiz e lista di Quiz (quiz.h/quiz.c)

Definire una struttura che contenga i dati relativi ad un quiz (vedere sopra) e definire, tramite questa, il tipo di nome **Quiz**. Definire una funzione di nome **readQuizListFromTxt** che prenda in ingresso un puntatore a file e restituisca una lista di strutture dati di tipo **Quiz** (la lista può essere implementata tramite puntatori o utilizzando l'ADT lista) – allo scopo di fattorizzare meglio il codice, si consiglia di effettuare la

Laboratorio di Informatica L-A

Prova d'Esame 3 – 9 Aprile 2008

lettura del singolo **Quiz** all'interno di un'ulteriore funzione opportunamente definita. Si tenga conto del fatto che il numero di quiz contenuti nel file non è noto a priori. Per un esempio di file di quiz, fare riferimento al file **Quiz.txt** contenuto nello start kit. La struttura del file è la seguente:

```
NumeroRisposte[int] ; Domanda[stringa]
Codice[char] ; Punteggio[float] ; Risposta[stringa]
Codice[char] ; Punteggio[float] ; Risposta[stringa]
Codice[char] ; Punteggio[float] ; Risposta[stringa]
...
```

Fra parentesi quadre compare il tipo di dato: l'informazione relativa al tipo non è contenuta realmente nel file.

Definire una funzione di nome **renderQuiz** che prenda in ingresso un quiz e ne stampi a console il testo e le risposte possibili (per la stampa delle risposte utilizzare **renderAnswer**).

Facoltativo: definire una funzione **destroyQuizList** che consenta di deallocare una lista di **Quiz**.

Nel **main**, si verifichi il funzionamento del codice scritto leggendo il file **Quiz.txt** e stampando a console tutti i quiz letti. Al termine del test commentare il codice senza eliminarlo.

Esercizio 3 – Esecuzione del quiz (quiz.h/quiz.c)

Definire una funzione di nome **readAnswerFromUser** che prenda in ingresso un **Quiz**, rimanga in attesa dell'input dell'utente (il codice di una risposta) e restituisca la risposta (**Answer**) corrispondente al codice preso in ingresso – la funzione deve terminare solamente se l'utente ha inserito un codice valido; se il codice non è valido occorre notificare l'utente dell'errore e rimanere in attesa di nuovo input. Definire una funzione di nome **doAssessment** che prenda in ingresso una lista di **Quiz**, mostri uno ad uno i quiz contenuti nella lista (usare **renderQuiz**), per ognuno di questi richieda l'inserimento del codice della risposta (usare **readAnswerFromUser**) e restituisca il punteggio totalizzato ottenendo sommando i punteggi delle singole risposte scelte dall'utente.

Nel **main**, si verifichi il funzionamento del codice scritto leggendo il file **Quiz.txt** ed "eseguendo" il quiz.