

# Laboratorio di Informatica L-A

## Prova d'Esame 2 – 17 Gennaio 2008

**Prima di cominciare:** si scarichi il file **StartKit2.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 software house che produce un'applicazione per la gestione di magazzino di farmacie deve produrre un'estensione al programma per ottimizzare le ricerche dei farmaci "originali" e dei corrispondenti "generici". L'estensione deve essere realizzata sotto forma di prototipo, deve essere in grado di leggere da file di testo l'elenco dei farmaci e di effettuare ricerche.

Un farmaco deve essere modellato come segue:

- nome del farmaco: al più 50 caratteri;
- indicazione del fatto che il farmaco sia originale o generico (un carattere o un booleano);
- principio attivo:
  - nome: al più 60 caratteri;
  - dosaggio (in mg): un intero;
- nome produttore: al più 50 caratteri;
- giacenza (numero di confezioni presenti in magazzino): un intero.

Le ricerche devono poter essere effettuate per:

- nome del farmaco: la ricerca deve essere effettuata per parola parziale, deve pertanto essere restituita una lista di farmaci il cui nome inizi per la stringa di ricerca data;
- principio attivo principale e dosaggio: recuperare tutti i farmaci che contengono il principio specificato nel dosaggio specificato – eventualmente deve essere possibile ignorare il dosaggio quindi recuperare tutti i farmaci contenenti un certo principio attivo a prescindere dal dosaggio dello stesso.

Gli elenchi di farmaci restituiti dalle ricerche devono essere opportunamente ordinati (vedere più avanti).

**Importante:** poiché è necessario implementare anche la deallocazione della memoria, è bene non utilizzare direttamente le funzioni dell'ADT lista che, per costruzione, non permette un corretto controllo della memoria allocata.

### *Esercizio 1 – Definizione della struttura Farmaco (farmaco.h/farmaco.c)*

Definire una struttura atta a contenere i dati relativi ad un principio attivo (nome, dosaggio; per le dimensioni vedere sopra) e definire, tramite questa il tipo di nome **PrincipioAttivo**. Su tale tipo definire una funzione di confronto di nome **comparePrincipioAttivo** che prenda in ingresso due principi attivi e un valore booleano (definire opportunamente il tipo **Boolean** in un apposito *header file*) e restituisca un intero; il valore booleano indica se nel confronto vada considerato o meno il dosaggio del principio attivo (confronto per nome + dosaggio o solo per nome), il valore intero restituito deve avere la stessa semantica dell'intero restituito dalla funzione **strcmp**. La funzione deve consentire di ordinare i principi attivi prima per nome (ordine alfabetico crescente), poi (eventualmente) per dosaggio.

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

- Creazione/Distruzione: la funzione **createFarmaco** deve allocare in memoria dinamica lo spazio necessario per ospitare il tipo **Farmaco**, inizializzare a valori nulli (a seconda del tipo) i campi della struttura e restituire un puntatore all'area di memoria allocata; la funzione **destroyFarmaco** deve deallocare la memoria allocata per una struttura farmaco il cui puntatore sia stato restituito da **createFarmaco**.
- Confronto: la funzione **compareFarmaco** deve prendere in ingresso due puntatori a **Farmaco** e restituire un intero con la stessa semantica dell'intero restituito dalla funzione **strcmp**. Il confronto

## Laboratorio di Informatica L-A

### Prova d'Esame 2 – 17 Gennaio 2008

deve consentire di ordinare i farmaci prima per principio attivo (tenendo conto anche del dosaggio – usare la funzione **comparePrincipioAttivo** precedentemente definita), poi per tipo (prima gli originali poi i generici) ed infine per nome (ordine alfabetico crescente).

- Lettura da file di testo/scrittura su *console*: la funzione **readFarmacoFromTxt** deve prendere in ingresso un puntatore a file e restituire un puntatore a **Farmaco** nel caso in cui la lettura sia stata fatta con successo, **NULL** altrimenti; la funzione **printFarmaco** deve prendere in ingresso un puntatore a **Farmaco** e stamparlo su *console*.

Nel **main**, si verifichi il funzionamento del codice scritto con opportune chiamate alle funzioni costruite. Al termine del test commentare il codice senza eliminarlo.

#### Esercizio 2 – Lista di farmaci (*farmacoList.h/farmacoList.c*)

Definire un ADT lista di puntatori a **Farmaco** denominato **FarmacoList**. Sul tipo **FarmacoList** devono essere implementate le operazioni che seguono:

- Aggiunta di un elemento ad una lista: la funzione **addToFarmacoList** deve prendere in ingresso un **FarmacoList** ed un puntatore a **Farmaco** e deve inserire tale puntatore alla lista in modo ordinato (sfruttare la funzione **compareFarmaco** di cui all'esercizio precedente). La funzione deve restituire il **FarmacoList** elaborato.
- Distruzione di un **FarmacoList**: la funzione **destroyFarmacoList** deve prendere in ingresso un **FarmacoList** e un booleano indicante se si debba deallocare solo la lista oppure anche i farmaci (**Farmaco\***) in essa contenuti.
- Lettura di un **FarmacoList** da file/scrittura su *console*: la funzione **readFarmacoListFromTxt** deve prendere in ingresso un puntatore a file aperto su un file di testo e restituire un **FarmacoList** contenente tutti i farmaci presenti nel file; la funzione **printFarmacoList** deve prendere in ingresso un **FarmacoList** e stampare il suo contenuto su *console*.

Nel **main**, si verifichi il funzionamento del codice scritto con opportune chiamate alle funzioni costruite. Al termine del test commentare il codice senza eliminarlo.

#### Esercizio 3 – Ricerche per nome parziale, principio attivo (*farmaco.h/farmaco.c, farmacoList.h/farmacoList.c*)

Allo scopo di implementare la ricerca per nome parziale del farmaco, definire una funzione di nome **farmacoNomeStartsWith** (*farmaco.h/farmaco.c*) che prenda in ingresso un puntatore a **Farmaco** e una stringa (di opportuna dimensione) e che restituisca un valore booleano che indichi se il nome del farmaco comincia per la stringa data – il confronto non deve essere case sensitive. A tale scopo, prima di fare il confronto occorre convertire in *upper case* entrambe le stringhe (il nome del farmaco e la stringa data) – la conversione delle stringhe deve avvenire carattere per carattere utilizzando la funzione **toupper** (contenuta nell'*header file ctype.h*) che prende in ingresso un carattere e restituisce il corrispondente maiuscolo (se disponibile). La funzione di ricerca per nome parziale deve essere denominata **findByPartialNome** (*farmacoList.h/farmacoList.c*), deve prendere in ingresso una lista di farmaci ed una stringa di dimensione opportuna (la stringa da cercare) deve restituire una nuova lista contenente i soli farmaci il cui nome comincia per la stringa data; tale verifica deve basarsi sulla funzione **farmacoNomeStartsWith** precedentemente scritta.

La funzione di ricerca per principio attivo deve essere denominata **findByPrincipioAttivo**, deve prendere in ingresso una lista di farmaci, un **PrincipioAttivo** e un booleano indicante se ignorare o meno l'indicazione di dosaggio presente nel principio attivo; la funzione deve restituire una lista di farmaci che contengano il principio attivo indicato, eventualmente ignorando l'indicazione di dosaggio.

Nel **main**, si verifichi il funzionamento del codice scritto leggendo il file di testo fornito nello *start kit* ed effettuando una ricerca per nome parziale, una ricerca per principio attivo con indicazione di dosaggio, una ricerca per principio attivo senza indicazione di dosaggio; in tutti e tre i casi, stampare il risultato della ricerca.

Laboratorio di Informatica L-A  
Prova d'Esame 2 – 17 Gennaio 2008