Übungen zur Vorlesung
„Grundlagen der Informatik 1“
Studiengang E-Technik

WS 2001/02

AG Betriebssysteme, FB 3
Jan Bredereke

ausgegeben: 25.1.2002
Abgabe und Besprechung:
in der Übung am **31.1.2002**

Serie 6

Suchen

Aufgabe 1: Suchen einer Rechnung

In der Vorlesung am 18.1. ist eine Bibliotheksfunktion zur binären Suche vorgestellt worden. Außerdem wurde eine Anwendung vorgestellt, die diese Bibliotheksfunktion nutzt und eine Namens-Gehalts-Liste durchsucht.

Schreibt eine andere Anwendung, das dieselbe Bibliotheksfunktion nutzt, aber eine Liste von Rechnungsnummern durchsucht. Rechnungsnummern sind dabei keine Strings, sondern *Integers* zwischen 10000 und 99999. Zu einer Rechnungsnummer gehören ein Datum, ein Euro-Betrag und ein Empfänger. Dies sind jeweils Strings. Definiert wie in der obigen Anwendung eine feste, vorsortierte Tabelle mit den folgenden Einträgen:

Rechnungsnummer	**Datum**	**Betrag**	**Empfänger**
10003	4.1.2002	27,53	M. Meyer
10004	4.1.2002	200,00	S. Schulze
33333	5.1.2002	22,03	S. Schmidt
40735	10.1.2002	94,53	T. Tietjen
40736	11.1.2002	1,00	S. Sparsam
42420	13.1.2002	87,32	M. Mueller
43001	14.1.2002	33,07	L. Lehmann

Definiert euch geeignete Datenstrukturen und eine geeignete Vergleichsfunktion `compareTo()`. Entsprechend wie in der obigen Anwendung soll eine Rechnungsnummer erfragt werden, nach ihr gesucht werden und gegebenenfalls der gefundene Rechnungs-Datensatz ausgegeben werden.

Die Bibliotheksfunktion ist im WWW zu finden unter
„`http://www.tzi.de/agbs/lehre/ws0102/gdi1/hintergrund-info/binsearch.c.txt`“ und
„`http://www.tzi.de/agbs/lehre/ws0102/gdi1/hintergrund-info/binsearch.h.txt`“.
Die Anwendung aus der Vorlesung steht unter
„`http://www.tzi.de/agbs/lehre/ws0102/gdi1/hintergrund-info/binaere_suche2.c.txt`“.

Hinweis: Zum korrekten Lösen der Aufgabe ist es nützlich, beim Übersetzen möglichst strenge Typprüfungen durchführen zu lassen. Übersetzt daher am besten mit `cc -Wall ...`, um alle Warnungen anzeigen zu lassen.

Zum Schreiben eurer Anwendung müßt ihr euch zumindest an die Schnittstellendefinition in `binsearch.h` halten. Daher ist diese Datei auf der nächsten Seite noch einmal abgedruckt.

Probiert euere Anwendung an Beispielen aus.

```
/*=========================================================================
 * wiederverwendbare bibliothek fuer binäre suche
 *=========================================================================*/

/*-- Anwender muss einen Array zur Verfuegung stellen,
     dessen Elemente in der ersten Komponente auf
     den Schluesselwert verweisen und in der 2. Komponente
     auf die Nutzdaten. Die tatsaechliche Struktur
     der Nutzdaten ist fuer die binaere Suche irrelevant.
     --*/
typedef struct {
    void *key; /* Zeiger auf Schluessel */
    void *data; /* Zeiger auf Nutzdaten */
} my_array_t;




/*
Die Anwendung ruft Bibliotheksfunktion binSearch() auf:

 theKey: Zeiger auf den Schluesselwert des gesuchten
         Datensatzes.
 a : Zeiger auf den Beginn des Arrays.
 len : Anzahl der Array-Elemente

 */
extern void *binSearch(void           *theKey,
                       my_array_t     *a,
                       unsigned int   len );


/*
Die Anwendung muss eine Vergleichsfunktion compareTo()
zur Verfuegung stellen, welche die ==, <, > Vergleiche auf
dem Datentyp der verwendeten Schluessel durchfuehrt.

Sie liefert einen beliebigen Wert < 0 zurueck,
wenn k1 in dieser Vergleichsrelation kleiner als k2 ist,
einen beliebigen Wert > 0, falls k1 > k2 und
0 bei Gleichheit beider Schluessel.
*/
extern int compareTo(void *k1, void *k2);
```