

---

## 3. Übungsblatt

### 7. Aufgabe

Gegeben sei das nachfolgende Programm. Geben Sie die Ausgaben des Programms an.

```
#include <stdio.h>

struct {
    int *pElement1;
    int *pElement2; } Element1, Elemente2;

main()
{
    int i, k=0, *j;
    int eFeld[3][3];

    for (i=2; i>=0; --i)
        for (j=eFeld[i]+2; j>=eFeld[i]; --j, ++k)
            *j = i+k;

    printf("%d %d %d %d\n",
           eFeld[0][0],eFeld[1][1],eFeld[2][2], **eFeld);

    Element1.pElement1 = eFeld[0];
    Element1.pElement2 = eFeld[1];
    Elemente2= Element1;
    Elemente2.pElement2 = &eFeld[2][0];

    printf("%d %d %d\n", Element1.pElement1[1],
           Elemente2.pElement1[0], Elemente2.pElement2[2]);

    for (i=0; i<6; ++i)
        Element1.pElement2[i] = Elemente2.pElement1[i]+i;

    printf("%d %d %d\n", Element1.pElement1[2],
           Element1.pElement2[2], Elemente2.pElement2[2]);
}
```

---

## 8. Aufgabe (Praktikum)

Implementieren Sie einen Teil einer Lagerverwaltung. In dem Lager bestehen Artikel aus folgenden Daten:

- Artikelnummer: Eindeutige Nummer im Bereich 1-200.000
- Artikelname: Name mit maximal 20 Zeichen
- Bestand: Anzahl der verfügbaren Artikel (integer-Wert)

Gehen Sie davon aus das eine Datei **Artikel.txt** existiert, in der alle Artikeldaten (unsortiert) abgelegt sind. In dieser Datei sollen je Zeile die oben genannten Daten je eines Artikels stehen. Da nicht bekannt ist wie viele Artikel-Daten in der Datei existieren, aber ein Artikeldatensatz sehr schnell über seine Artikelnummer selektierbar sein soll, müssen Sie Ihr Programm zum Suchen eines Artikels optimieren. Deshalb sollen alle Artikeldaten in einem binären Baum gehalten werden. Die Daten werden hierbei bezüglich der Artikelnummer in den Baum einsortiert.

Schreiben Sie ein Programm mit folgenden Eigenschaften und Funktionalitäten:

- Beim Start Ihres Programms werden die Daten der Datei **Artikel.txt** eingelesen und in einen Binären Baum einsortiert
- Nach dem Einlesen kann sich der Benutzer einen Artikel am Bildschirm anzeigen lassen. Hierzu gibt er die Artikelnummer ein. Das Programm sucht dann den Datensatz und zeigt diesen (so er vorhanden ist) an.
- Für einen angezeigten Artikel kann der Benutzer den Bestand ändern. Der binäre Baum wird entsprechend aktualisiert.

**Wichtig:** Implementieren Sie Ihr Programm modular. Trennen Sie die Baum basierte Datenhaltung vom Hauptprogramm, indem Sie ein separates Modul erstellen. Dieses soll folgende Funktionalitäten zur Verfügung stellen:

- Datentyp: ArtikelBaum
- Prozedur: ArtikelEinfuegen
- Funktion: ArtikelSuchen
- Prozedur: ArtikelAktualisieren
- Prozedur: ArtikelLoeschen (**optional**)

PS.: Wer möchte/kann, kann diese Aufgabe auch objekt-orientiert in C++ lösen.

---

## 9. Aufgabe

Programmieren Sie einen Programm, das ganze Zahlen von einer Datei einliest und in aufsteigender Reihenfolge sortiert wieder ausgibt.

Realisieren Sie das Programm so, daß

- a) die eingelesenen Zahlen in einem Feld sortiert werden. In diesem Fall gibt die erste Zahl die Anzahl der einzulesenden Zahlen an!
- b) in einer einfach verketteten Liste mit der größten Zahl am Anfang der Liste
- c) in einer doppelt verketteten Liste mit der kleinsten Zahl am Anfang
- d) in einem binären Baum sortiert werden.