

## Prozedurale Programmierung

### Übungsaufgabe zu Funktionen mit Zeigern im Integer-Array

#### Aufgabenstellung:

Schreiben Sie ein Programm mit folgender Funktionalität:

- Im Hauptprogramm soll ein `int`-Array mit **1001** Elementen für Zufallszahlen definiert werden.
- Arbeiten Sie innerhalb der folgenden Funktionen mit Zeigerarithmetik! (siehe Hinweis am Ende der Aufgabenbeschreibung). Der Zugriff mit Subskription auf ein Array in der Form **`myarray[i]`** ist nicht erlaubt!
- Erstellen Sie eine Funktion **`init()`**, die einen `int`-Zeiger und die Arraygröße (Anzahl der Elemente) entgegen nimmt. Befüllen Sie das Array über den Zeiger mit Zufallszahlen zwischen 1 und **9999**.
- Zunächst wird der Zufallszahlengenerator durch einmaligen Aufruf der Funktion **`srand(time(NULL))`** gestartet. Danach liefert jeder Aufruf von **`rand()`** eine Zufallszahl. Diese müssen Sie in den Zahlenbereich von 1 bis 9999 umwandeln. Suchen Sie die notwendigen include-Anweisungen für **`srand()`**, **`rand()`** und **`time()`** heraus.
- Schreiben Sie eine Funktion **`ausgabe()`**, die einen `int`-Zeiger auf das Array und die Anzahl der Elemente entgegen nimmt und den Inhalt des Arrays formatiert ausgibt (5 Stellen pro Zahl).  
Bsp.:      2001   7347   6151      56   ...  
              7      42   2293   9417   ...
- Rufen Sie im Hauptprogramm die Funktionen **`init()`** und **`ausgabe()`** für das Array auf.
- Erstellen Sie eine Funktion **`tausche()`**, die zwei `int`-Zeiger entgegen nimmt und den Inhalt der beiden Variablen, auf die die Zeiger zeigen, vertauscht.
- Schreiben Sie eine Funktion **`sortiere()`**, die die Reihenfolge aller Array-Elemente im Array so vertauscht, dass sie der Größe der Zahlen nach sortiert sind, die niedrigsten am Anfang, die höchsten am Ende. Benutzen Sie dazu den Bubblesort-Algorithmus (siehe Hinweis am Ende) und die Funktion **`tausche()`**. Bestimmen Sie die benötigten Parameter für die Funktion.
- Übergeben Sie im Hauptprogramm das Array an die Funktion **`sortiere()`** und nochmals an **`ausgabe()`**.

- Rufen Sie im Hauptprogramm die Funktionen ***init()*** und ***ausgabe()*** für das Array ein zweites Mal auf.
- Schreiben Sie eine Funktion ***int \* mini()***, die einen Zeiger auf den kleinsten Wert innerhalb des Arrays bestimmt und zurückgibt. Verwenden Sie die Zeigerarithmetik und eine Schleife.
- Schreiben Sie eine Funktion ***int \* maxi()***, die einen Zeiger auf den größten Wert innerhalb des Arrays bestimmt und zurückgibt. Verwenden Sie die Zeigerarithmetik und eine Schleife.
- Schreiben Sie eine Funktion ***int \* median()***, die einen Zeiger auf den Median Wert innerhalb des Arrays bestimmt und zurückgibt. Tipp: Suchen Sie den Median erst nach der erneuten Sortierung. Schauen Sie in der Literatur nach, was der Median ist. Vielleicht ist die Lösung nicht schwierig.
- Die drei Zeiger sollen in der Funktion bestimmt werden, die Ausgabe soll im Hauptprogramm sein. Beachten Sie, dass Sie den zurückgegebenen Zeiger dereferenzieren müssen.

Das Programm muss fehlerfrei und ohne Warnungen mit der Compileroption **-Wall** compilieren.

Es dürfen keine globalen Variablen verwendet werden.

Schreiben Sie Ihren Namen in den Kopfkomentar und kommentieren Sie Ihr Programm an den wichtigen Stellen.

---

Hinweis:

Der Bubblesort-Algorithmus (mit Pointer-Arithmetik):

```
// Bubble Sort Algorithmus (unoptimiert)
// p ist der Zeiger auf das Array, n und i sind int Laufvariablen

for (n = anzahl; n>1; n--) {
    for (i = 0; i<n-1; i++) {
        if (*(p+i) > *(p+i+1)) {
            tausche(p+i, p+i+1);
        }
    }
}
```