

Programmiersprachen (C++)

Übung 3 Anweisungen, Felder, Strukturen

Aufgabe 1

Eine Wetterbeobachtungsstation hat über zehn Jahre hinweg die jährlichen Niederschlagsmengen gemessen und in einem Feld notiert:

```
double niederschlag[] = {69.1, 50.1, 67.0, 63.0, 76.5,  
54.1, 86.6, 55.9, 75.7, 57.2};
```

Schreiben Sie ein C++-Programm, das die durchschnittliche jährliche Niederschlagsmenge berechnet und ausgibt. Außerdem soll eine Tabelle ausgegeben werden, die für jedes Jahr die Niederschlagsmenge und die jeweilige Abweichung vom Mittelwert enthält.

Die Ausgabe Ihres Programms könnte zum Beispiel folgendermaßen aussehen:

```
Die durchschnittliche Niederschlagsmenge betraegt 65.52 cm.
```

Jahr	Niederschlag	Abweichung vom Durchschnitt
1	69.10	3.58 cm ueber dem Mittelwert
2	50.10	15.42 cm unter dem Mittelwert
3	67.00	1.48 cm ueber dem Mittelwert
4	63.00	2.52 cm unter dem Mittelwert
5	76.50	10.98 cm ueber dem Mittelwert
6	54.10	11.42 cm unter dem Mittelwert
7	86.60	21.08 cm ueber dem Mittelwert
8	55.90	9.62 cm unter dem Mittelwert
9	75.70	10.18 cm ueber dem Mittelwert
10	57.20	8.32 cm unter dem Mittelwert

Zur Verwendung der folgenden, so genannten Manipulatoren müssen Sie die Deklarationsdatei iomanip.h inkludieren:

```
#include <iomanip>
```

Die Anweisung

```
cout << fixed << setprecision(2);
```

bewirkt, dass Gleitkommawerte in Festpunktnotation mit zwei Nachkommastellen ausgegeben werden.

Die Anweisung

```
cout << setw(12) << a;
```

sorgt dafür, dass der Wert der Variablen a mit einer Gesamtweite von mindestens zwölf Zeichen ausgegeben wird. Bei Bedarf werden links Leerzeichen eingefügt, um die angegebene Gesamtweite zu erreichen.

Aufgabe 2

Die folgende Implementierung des Sortierverfahrens Bubble Sort hat den Nachteil, dass die äußere Schleife auch dann n -mal durchlaufen wird, wenn das übergebene Feld bereits sortiert ist. Die äußere Schleife sollte abgebrochen werden, wenn festgestellt wurde, dass das Feld korrekt sortiert ist. Das Feld ist sortiert, wenn die äußere Schleife einmal durchlaufen wurde, ohne dass eine Vertauschung vorgenommen wurde. Verändern Sie die folgende - in der Datei

V:\kuenkler\CPP\uebungen\ueb03\BubbleSort.cpp

enthaltene - Funktion bubble entsprechend.

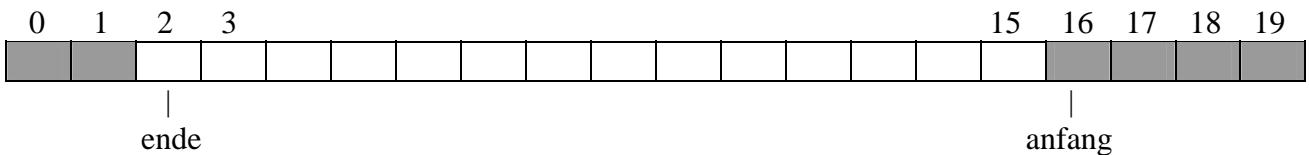
```
void bubble(int x[], int len)
{
    for(int i = len-1; i > 0; i--)
        for(int j = 0; j < i; j++)
            if(x[j] > x[j+1])
            {
                int tmp = x[j];
                x[j] = x[j+1];
                x[j+1] = tmp;
            }
}
```

Aufgabe 3

Es sei die folgende Definition einer Warteschlange (Queue) q gegeben, in der int-Werte gespeichert werden können:

```
const int MAX = 20;
struct queue {
    int anfang, ende, laenge;
    int daten[MAX];
} q = {0, 0, 0,};
```

Die Warteschlange soll als Feld implementiert werden, das - wie im folgenden Beispiel dargestellt - als Ring betrachtet wird. Das Feld daten hat 20 Elemente; die grau schattierten Elemente seien belegt.



Schreiben Sie die folgenden Funktionen:

```
void enqueue(int elem)    // Einfügen eines Elementes in die Queue
int dequeue()             // Entnehmen eines Elements aus der Queue
int anzele()              // Anzahl der in der Queue enthaltenen Elemente
int sumele()              // Summe der in der Queue enthaltenen Elemente
```

Testen Sie Ihre Funktionen sorgfältig.