

Kompaktkurs Einführung in die Programmierung

Übungsblatt 7: Felder

Lernziele

- Initialisieren und Umgang mit (eindimensionalen) Feldern (Arrays).
- Umgang mit Zeichenketten.
- Sortierung von Feldern.

1) Felder zum Üben (T)

Was ist nach Ausführung der folgenden Anweisungen jeweils der Inhalt des Feldes `a`?

i) Normale Indizes:

```
int a[10], i, j;
for (i=0; i<10; i++) {
    a[i] = 0;
    for (j=0; j<i; j++) a[i] = a[i] + 1;
}
```

ii) Feldkomponenten als Indizes:

```
int a[10], i;
a[0] = 1;
for (i=0; i<9; i++) a[a[i]] = a[i]+1;
```

2) Vektornorm (P)

Schreiben Sie eine Funktion, die die euklidische Norm eines Vektors berechnet. Der Vektor ist ein Feld von `double`-Werten. Zusätzlich übergeben wir die Zahl der Komponenten (also eine nichtnegative ganze Zahl), so daß sich folgende Signatur ergibt: `double norm(double *x, int n)`

Implementieren Sie diese Funktion und testen Sie sie durch geeignete Funktionsaufrufe. Die Berechnung von Wurzeln ermöglicht Ihnen (wie in Aufgabe 4 von Blatt 4) der C-Header `<math.h>` (in C++ `<cmath>`) über die Funktion `double sqrt(double d)` (ggf. beim Übersetzen die Option `-lm` hinter dem Dateinamen angeben).

3) Palindrome (P)

Schreiben Sie ein Programm `palindrom.cpp`, das überprüft, ob eine Zeichenkette ein Palindrom ist (rückwärts gelesen dasselbe Wort ergibt, z.B. "gnubelebung"). Dieses könnte eine Funktion der Signatur `bool palindrom(char *s)` haben die als Rückgabewert `true` liefert, falls `s` ein Palindrom ist, andernfalls `false`. Die leere Zeichenkette sei ebenfalls ein Palindrom.

Hinweise:

- Über den C-Header `#include <ctype.h>` (in C++ `#include <cctype>`) steht Ihnen die Funktion `int tolower(int c)` zur Verfügung die ein einzelnes Zeichen ggf. in den entsprechenden Kleinbuchstaben umwandelt. Damit lässt sich bewirken, dass auch groß geschriebene Wörter (z.B. "Gnubelebung") als Palindrome erkannt werden.

4) Sortierung mit Bubblesort (P)

Sortierung von Daten ist wesentlicher Bestandteil der Datenverarbeitung. Implementieren Sie eine Funktion `bubblesort()` die ein beliebig langes Feld aus `int`-Werten entgegennimmt und dieses aufsteigend sortiert. Demonstrieren Sie ihre Implementierung durch eine Ausgabe des Feldes vor und nach der Sortierung.

Bubblesort iteriert über jedes Element und vertauscht Nachbarn wenn das linke Element größer ist als das Rechte. In erster Iteration wird damit das größte Element ans Ende sortiert.

Beispiel am Feld [5 3 4 2 1]:

```
----- Iteration 1
[ 5 3 4 2 1 ]
[ 3 5 4 2 1 ]
[ 3 4 5 2 1 ]
[ 3 4 2 5 1 ]
[ 3 4 2 1 5 ]
----- Iteration 2
[ 3 4 2 1 5 ]
[ 3 2 4 1 5 ]
[ 3 2 1 4 5 ]
----- Iteration 3
[ 2 3 1 4 5 ]
[ 2 1 3 4 5 ]
----- Iteration 4
[ 1 2 3 4 5 ]
```

Achten Sie dabei darauf, dass Sie den Algorithmus effizienter gestalten:

- Das Feld muss pro Iteration nicht jedesmal komplett durchlaufen werden.
- Wie gehen Sie mit Feldern um die bereits sortiert vorliegen?

Ihre Implementierung sollte dabei auch Felder sortieren die Duplikate, Nullen sowie negative Zahlen enthalten.

5) Zeiger auf Funktionen (★)

Ziel dieser Aufgabe ist es einen Mini-Taschenrechner mit Hilfe von Zeigern auf Funktionen zu realisieren.

- i) Machen Sie sich mit Zeigern die auf Funktionen zeigen sowie mit Arrays die Funktionszeiger beinhalten können vertraut.
- ii) Realisieren Sie eine Eingabe, so dass der Nutzer den Operator (+, -, *, /) und 2 Zahlen eingeben soll. Das Endergebnis der eingegebenen Operation soll Ihr Programm zum Ende auf die Konsole ausgeben.
- iii) Implementieren Sie die 4 Funktionen `add`, `sub`, `mul` und `div`, die jede 2 Gleitkommazahlen entgegennehmen und das Ergebnis entsprechend zurückgeben.
- iv) Realisieren Sie nun mit Hilfe eines Funktionszeigerarrays Ihren Mini-Taschenrechner.