

AIK Programmieren 3 Übung:

Referenz-Parameter, Wh Strukturen & Funktionen

Klaus Kusche

Sortieren

Gesucht ist ein Programm, das die Worte auf der Befehlszeile in sortierter Reihenfolge (gemäß **strcmp**) zeilenweise ausgibt. Dabei soll vor jedem Wort seine ursprüngliche Position (“Das wievielte Wort war das auf der Befehlszeile?”) ausgegeben werden.

Daher ist jedes zu sortierende Element eine Struktur bestehend aus einem Pointer auf den Text des Wortes (d.h. kein **char**-Array mit einer Kopie des Wortes, sondern ein **char *** mit einer Kopie des Pointers aus `argv`) und der Nummer des Wortes.

Sortiert wird ein Array solcher Strukturen; du darfst dessen Größe (max. Anzahl der Elemente) fix festlegen.

Das Hauptprogramm hat daher folgende Struktur:

- Prüfen der Befehlszeile: Gar keine oder zu viele Worte?
- Speichern der Worte samt Position im Array.
- Sortieren des Arrays.
- Ausgeben des sortierten Arrays.

Diesmal verwenden wir ein noch einfacheres Sortierverfahren als Bubble-Sort:

- Vergleiche jedes einzelne Element (beginnend mit dem ersten bis zum vorletzten) der Reihe nach mit allen Elementen dahinter (das werden zwei geschachtelte **for**-Schleifen, von wo bis wo läuft die zweite?).
- Wenn die zwei verglichenen Elemente in der falschen Reihenfolge stehen: Vertausche sie!

Das Vergleichen und das Vertauschen soll mit zwei eigenen Funktionen geschehen:

- **verkehrt** wird mit zwei Elementen (Strukturen) aufgerufen und liefert **true**, wenn der Text des ersten Elementes größer als der des zweiten ist, sonst **false**.
- **tausche** wird mit zwei Elementen (Strukturen) aufgerufen und vertauscht deren Inhalt (erinnere dich: Man kann Strukturen als Ganzes zuweisen, es ist nicht nötig, die Struktur-Member einzeln zu kopieren). Diese Funktion hat kein Ergebnis.

Schreib zwei Varianten des Programms mit verschiedener Übergabe der Struktur-Parameter an die Funktionen:

- Einmal wie in C üblich mit Pointern (so wie wir das voriges Jahr gemacht haben).
- Einmal als C++-Referenzparameter (&-Parameter)!

Bei der Funktion **verkehrt** könntest du auch die dritte Möglichkeit zur Parameter-Übergabe (“by Value”) probieren. Warum geht das bei **tausche** nicht?

Verwende auch C++ Ausgabe: **iostream** und **cout** << ...