**04 Woerter**

**Zusammenfassung der Problemstellung**

Das Programm soll ine Anzahl Wörter von der Tastatur einliest, diese in einem Array ablegt und zum Schluss alle Wörter alphabetisch sortiert ausgibt. Wiederholt eingegebene Wörter werden ignoriert und das Ende der Eingabe wird durch die Eingabe von „ZZZ“ markiert, d.h. danach werden die Wörter ausgegeben.

**Lösungsvarianten**

Die eingelesenen Wörter werden in einen Array von dynamisch allozierter Grösse gespeichert. Im Vergleich zu einem Array fixer Grösse sparen wir damit Platz.

In einer Liste speichern wir die Adressen aller eingelesenen Wörter. Der Vorteil daran ist, dass beim Sortieren keine Wörter sondern lediglich deren Pointer in der Liste geändert werden müssen.

Als Sortieralgorithmus verwenden wir Bubblesort. Da die Datenmenge relativ klein ist, fällt die Laufzeit nicht stark ins Gewicht und daher kann eine einfache Implementierung wie jene von Bubblesort verwendet werden.

**Modulübersicht**

Da es sich um ein sehr kleines Programm handelt, befindet sich der ganze Code in main.c, aufgeteilt auf die 4 Funktionen main, save\_to\_list, sort, print\_list.

**Globale Variablen:**

* char word[20]
* char \*wordlist[100]
* int anzworte
* int wortzaehler

**Lokale Variablen:**

* char \*wkopie (main)
* size\_t listlaenge (main)
* int z (sort)
* char \*temp (sort)
* int k (print\_list)

**Programmablauf**

Main liest eine Anzahl von Wörtern ein, die Methode save\_to\_list speichert die Adresse der Worte in der Liste und sortiert sie anschliessend mit der Methode sort. Zum Schluss wird die geordnete Liste von der Methode print\_list ausgegeben.

**Tests**

*test\_main\_ausgabe:* prüft, ob eine Anzahl eingegebene Wörter korrekt ausgegeben wird.

*test\_sort\_two\_equal\_word:* prüft, ob zwei gleiche Wörter nur einmal gespeichert werden

*test\_sort\_empty\_word*: prüft, ob eine leere Eingabe richtig behandelt wird

*test\_save:* prüft, ob die Adresse eines eingegebenen Wortes korrekt in der Liste gespeichert wird.

Alle Tests liefen erfolgreich durch.

**Erkenntnisse**

Das Programm läuft fehlerfrei aussert es werden mehr als zwei gleiche Worte eingegeben.

Mögliche Erweiterungen: Anzahl vorkommen der Wörter zählen.

**Anhang A: Quelltext**