

## Aufgabe 2: wsort (12.0 Punkte)

### a) Standard-Eingabe sortieren

Schreiben Sie ein Programm `wsort`, welches eine Liste von Wörtern vom Standard-Eingabekanal (`stdin`) einliest, diese Liste alphabetisch sortiert und die sortierte Liste auf dem Standard-Ausgabekanal (`stdout`) ausgibt.

Sowohl beim Einlesen als auch beim Ausgeben der Wörter steht jedes Wort in einer eigenen Zeile. Ein Wort umfasst **alle** Zeichen einer Zeile. Zeilen sind durch ein *Newline*-Zeichen (`\n`) voneinander getrennt, die selbst nicht Teil des Wortes sind. Bis auf die letzte Zeile enden alle Zeilen immer mit einem Newline-Zeichen. Die letzte Zeile muss kein Newline-Zeichen enthalten.

Wörter, die eine maximale Länge von 100 Zeichen überschreiten, werden mit einer entsprechenden Fehlermeldung ignoriert. Leere Zeilen sind ohne Fehlermeldung zu ignorieren.

Im Verzeichnis `/proj/i4sp1/pub/aufgabe2` finden Sie Beispiel-Eingabedateien (`wlist*`) sowie eine Vergleichimplemen-tierung (`wsort`), mit der Sie die Wortlisten sortieren und die Ausgabe jeweils mittels `diff(1)` mit der Ausgabe Ihres eigenen `wsort`-Programms vergleichen können.

Selbst wenn alle Beispiel-Eingabedateien von Ihrer Lösung korrekt sortiert werden, können dennoch Fehler in Ihrer Imple-mentierung enthalten sein. Verlassen Sie sich daher beim Testen nicht alleine auf die Beispiel-Eingabedateien.

Mit Hilfe von `malloc(3)` und `realloc(3)` können Sie dynamisch Speicher an- und nachfordern, um die benötigten Daten-strukturen anzulegen bzw. zu erweitern.

### b) Makefile schreiben

Erstellen Sie ein zur Aufgabe passendes Makefile, welches die beiden Targets `wsort` und `wsort-debug` unterstützt. Beide Targets übersetzen die Datei `wsort.c` und erzeugen eine entsprechende ausführbare Datei. Das Target `wsort-debug` sorgt dafür, dass Debug-Symbole in der ausführbaren Datei enthalten sind.

Wird an `make(1)` kein Argument übergeben, so soll die ausführbare Datei `wsort` erzeugt werden.

### Hinweise zur Aufgabe:

- Erforderliche Dateien: `wsort.c`, Makefile
- Hilfreiche *Manual-Pages*: `ferror(3)`, `fgets(3)`, `getchar(3)`, `malloc(3)`, `qsort(3)`, `realloc(3)`, `strcmp(3)`
- Sämtliche Fehlermeldungen sollen auf dem Standardfehlerkanal (`stderr`) ausgegeben werden. Auf die Standardausgabe (`stdout`) soll ausschließlich die sortierte Wortliste ausgegeben werden.
- Beachten Sie, dass bei der Korrektur `valgrind` zum Einsatz kommen wird. Testen Sie daher Ihre Lösung mit `valgrind` auf eventuelle Speicherzugriffsfehler und beseitigen Sie diese.

### Hinweise zur Abgabe:

Bearbeitung: Zweiergruppen

Bearbeitungszeit: 7 Werkstage

Abgabezeit: 17:30 Uhr