

## Prüfungsleistung für das Modul C/C++ im SoSe 2020

Die Prüfungsleistung besteht aus zwei Teilen, einem C- und einem C++-Teil. Die Lösungen für beide Teile sind spätestens bis zum 31.07.2020 per Mail an mich einzusenden.

### Aufgabenstellung zum C-Teil

Zu implementieren ist Insertsort für den Datentyp `int` in Form einer Bibliotheksfunktion.

- Das Sortierprogramm ist so zu erweitern, dass die Anzahl der Schlüsselvergleiche und die Anzahl der Zuweisungen von Schlüsselwerten in einem Sortierdurchlauf gezählt und an den Aufrufer zurückgegeben werden.
- Die Anweisungen zum Zählen sollen durch bedingte Compilierung ein- und ausschaltbar sein.
- Mit ihrem Programm sind dann die Werte in der Datei „unsortiert.txt“ einzulesen und zu sortieren. Die Anzahl der Zuweisungen und Vergleiche ist auszugeben. Die sortierten Werte sind in die Datei „sortiert.txt“ zu schreiben. Das Format soll der Datei „unsortiert.txt“ entsprechen.

Hinweise:

1. Die erste Zahl in der Datei „unsortiert.txt“ gibt die Anzahl der folgenden Werte an.
  2. Diese Zahl darf nicht in die Sortierung einfließen.
  3. Zahlen können mehrfach auftreten.
- Es ist eine Funktion zu schreiben, die ein Array der Größe  $n$  (also einer beliebigen, durch den Aufrufer vorgegebenen Größe) mit Zufallszahlen füllt. Diese Zufallszahlen sollen folgenden Bedingungen erfüllen:
    1. Es dürfen nur die Zahlen von 0 bis  $n-1$  vorkommen
    2. Jede Zahl darf nur einmal im Array vorkommen
    3. Die Zahlen sollen zufällig im Array stehen
    4. Jeder Aufruf muss eine andere Anordnung liefern
  - Mit Hilfe dieser Funktion werden nun Arrays der Größen 1.000, 2.000, 3.000, ..., 10.000 erstellt und mit Insertsort sortiert. Der Vorgang ist für jede Array-Größe 100 x zu wiederholen. Aus diesen jeweils 100 Wiederholungen ist das jeweilige arithmetische Mittel der Anzahl der Schlüsselvergleiche und die Anzahl der Zuweisungen von Schlüsselwerten zu bestimmen. Die 2 x 11 Mittelwerte sind in der Datei „Ergebnisse.txt“ inkl. der zugehörigen Arraygrößen abzulegen.
  - Zu den 22 Mittelwerten soll mit Excel eine Grafik erstellt werden, die die Mittelwerte in Abhängigkeit von der Array-Größe zeigt. Zusätzlich ist mit Excel eine quadratische Näherungsfunktion zu bestimmen und in die Grafik zu integrieren. Die beiden Funktionsgleichungen sind anzugeben.
  - Es ist eine Konsolen-Oberfläche zu erstellen, die folgenden Möglichkeiten bietet:
    1. Sortieren eines Arrays, dessen Größe und Werte durch den Benutzer eingegeben werden können. Das sortierte Array und die Anzahl der Schlüsselvergleiche und die Anzahl der Zuweisungen von Schlüsselwerten ist in der Konsole auszugeben.
    2. Sortieren einer beliebigen Datei im Format der Datei „sortiert.txt“. Der Dateiname wird vom Anwender eingegeben. Das Ergebnis wird in eine Datei mit dem Zusatz „-sort“ ausgegeben. Die Ausgabe der Anzahl der Schlüsselvergleiche und die Anzahl der Zuweisungen von Schlüsselwerten erfolgt auf dem Bildschirm.
    3. Start des Sortierens der Arrays der Größen 1.000, 2.000, 3.000, ..., 10.000. Die Anzahl der jeweiligen Wiederholungen ist vom Anwender abzufragen.

Abzugeben sind:

- Der dokumentierte C-Code
- Die Datei „sortiert.txt“ mit den Angaben zu der Anzahl der Schlüsselvergleiche und der Anzahl der Zuweisungen von Schlüsselwerten
- Die Datei „Ergebnisse.txt“ (für 100 Wiederholungen)
- Die mit Excel erstellte Grafik
- Die Funktionsgleichungen der Näherungsfunktionen
- Ein dokumentiertes und von Hand erstelltes makefile

Hinweise:

- Programmieren, Kommentieren und Dokumentieren Sie Ihr Programm so, dass andere C-Entwickler sich leicht einarbeiten können. Berücksichtigen Sie dabei zwei Szenarien:
  - Nutzung als Bibliothek in einem anderen Programm
  - Weiterentwicklung und ggf. Fehlerbeseitigung
- Die Programme sollen sich ohne Änderungen mit dem gcc übersetzen lassen.
- Die Bibliothek mit Insertsort soll so realisiert werden, dass sie sich leicht in andere Programme einbinden lässt.
- Die Realisierung soll "C-typisch" sein (Aufteilung in .h/.c/...-Dateien, ...)

[Aufgabenstellung zum C++-Teil](#)

Folgt später.