

Name: \_\_\_\_\_ Aufwand in h: \_\_\_\_\_

Mat.Nr: \_\_\_\_\_ Punkte: \_\_\_\_\_

Übungsgruppe: \_\_\_\_\_ korrigiert: \_\_\_\_\_

### **Arbeiten mit dem GCC (24 Punkte)**

Machen Sie sich mit den grundlegenden Eigenschaften der Programmiersprache C vertraut. Entpacken Sie die für diese Übung bereit gestellten Dateien (*C-Dateien für Übung2*) in ein eigenes Verzeichnis und analysieren Sie den C-Code. Versuchen Sie die Dateien zu einem ausführbaren Programm *ctest* zu übersetzen, indem Sie alle Fehler und Warnungen beseitigen. Kommentieren Sie die beseitigten Fehler, Warnungen und Codeverbesserungen im Quellcode entsprechend!

Benutzen Sie zur Verbesserung des Codes die Fehler-Checkliste auf der Kommunikationsplattform.

Verwenden Sie zur Übersetzung der Dateien den gcc von der Kommandozeile aus und bearbeiten Sie die C-Dateien in einem einfachen Editor (z.B. notepad++.exe)!

Analysieren Sie weiters den Ablauf des Programmes, kommentieren Sie alle Funktionen um den Ablauf entsprechend erklären zu können und machen Sie das Programm lauffähig! Nutzen Sie dazu auch die zur Verfügung gestellten Unterlagen und die Linux-Manpages (*Manual Page - Handbuch-Seite*).

<http://de.wikipedia.org/wiki/Manpage>

Beantworten Sie weiters folgende Fragen:

- Warum sind einige Funktionen als `static` deklariert und andere nicht?
- Welches Programmier-Prinzip lässt sich mit Hilfe von `static`-Funktionen in C realisieren?

**Allgemeine Hinweise:** Legen Sie bei der Erstellung Ihrer Übung großen Wert auf eine **saubere Strukturierung** und auf eine **sorgfältige Ausarbeitung**! Dokumentieren Sie alle Schnittstellen und versehen Sie Ihre Algorithmen an entscheidenden Stellen ausführlich mit Kommentaren! Testen Sie ihre Implementierungen ausführlich! Geben Sie den **Testoutput** mit ab!