

Prüfungsvorleistung Datenstrukturen

Allgemeine Hinweise

Als Prüfungsvorleistung für die Klausur Datenstrukturen müssen Hausaufgaben gelöst werden.

Jede Hausaufgabe wird mit einer bestimmten Anzahl an Punkten (4-6) bewertet.

Sie müssen insgesamt mindestens 50% der möglichen Gesamtpunktzahl **aller** Hausaufgaben erreichen, um die Prüfungsvorleistung zu erbringen.

Falls Sie die Prüfungsvorleistung schon in einem vorherigen Semester erbracht haben, ist die Lösung der Hausaufgaben nicht nötig.

Alle Aufgaben müssen **individuell** gelöst werden. Nachweisbar identische bzw. sehr ähnliche Lösungen werden mit 0 Punkten bewertet.

Nach erfolgter Korrektur können die Ergebnisse inkl. kurzer Bewertungskommentare im Opal eingesehen werden.

Dokumente

Grafiken, Tabellen, textuelle Lösungen

Erstellen Sie immer ein **PDF-Dokument**, wenn Grafiken, Tabellen oder textuelle Lösungen gefordert sind.

Quelltexte

Ihre Lösungen sind in C bzw. C++ zu kodieren.

Nur selbsterstellte und **übersetzbare** Quelltexte sind abzugeben (ggf. mit Makefile), keine ausführbaren Programme oder kompilierte Objektdateien.

Quelltexte sind vernünftig zu dokumentieren.

Bibliotheken

Grundsätzlich dürfen Sie Bibliotheksfunktionen nur zur Realisierung von Ein- oder Ausgabefunktionalität (Tastatur- oder Dateieingabe, Bildschirm- oder Dateiausgabe) nutzen.

Abgabemodalitäten

Die Abgabe der Hausaufgaben erfolgt im OPAL. Dazu müssen Sie sich im Opal einschreiben:

<https://bildungsportal.sachsen.de/opal/url/RepositoryEntry/8419999746>

Der (späteste) Abgabetermin ist auf jeder Aufgabenstellung individuell vermerkt.

Beachten Sie bei der Abgabe folgende Konventionen:

- Legen Sie alle abzugebenden Dokumente (Quelltexte, Dokumentation, Sonstiges) in einem Verzeichnis ab, das Sie wie folgt benennen: "HA"<Nummer der Hausgabe>"_ "<Matrikelnummer>
z.B. HA2_123456
- Archivieren und Komprimieren Sie das Verzeichnis **ausschließlich** mittels ZIP (d.h. weder gzip noch bzip noch 7z noch zipx). Benennen Sie entstehende ZIP-Datei wie Ihr Verzeichnis plus der Extension ".zip", z.B. HA2_123456.zip
- Eine Abgabe Ihrer Lösung ist bis spätestens 23:59 Uhr des angegebenen Tages möglich.
- Eine Nichtbeachtung dieser Hinweise kann zu **maximalem Punktabzug** führen!

Kompatibilität

Um eine einheitliche Basis zur Korrektur und zum Test Ihrer Programme zu schaffen, wird die Lauffähigkeit Ihrer Lösungen auf einer einheitlichen Plattform vorausgesetzt.

Dazu ist es notwendig, dass Sie Ihre Quelltexte auf dem Login-Server der TU-Chemnitz testen (<https://login.tu-chemnitz.de>).

Der Login-Server bietet jedem URZ-Nutzer den Zugriff auf einen vorinstallierten C++ - Compiler (GCC), mit dem Quelltexte, die im AFS-Homeverzeichnis liegen, übersetzt werden können.

Vorausgesetzt werden Basiskenntnisse zur Bedienung einer UNIX-Kommandozeilenumgebung (Shell).

Lösungen von Programmieraufgaben, die **nicht** auf dem Login-Server übersetzt (und zum Laufen gebracht) werden können, werden **grundsätzlich mit 0 Punkten** bewertet.

Verwendung des Login-Servers

Anmeldung

```
ssh [URZ-Kürzel]@login.tu-chemnitz.de
```

Nach Eingabe Ihres Passworts haben Sie Zugriff auf die Dateien in Ihrem AFS-Homeverzeichnis, eine Übersicht erhalten Sie z.B. mit Eingabe des Kommandos `ls`

Übersetzen eines einfachen C++ - Programms

```
g++ -o [Dateiname des zu erzeugenden Programms] [Name der Quelltextdatei]
```

Beispiel (Übersetzen des Quellcodes einer Datei „test.cpp“)

```
g++ -o test test.cpp
```

Ausführen des erzeugten Programms:

```
./test
```

Parameterübergabe

Um Ihre Quelltexte mit verschiedenen Eingabedaten testen zu können, ist es nötig, diese als Dateien bereitzustellen.

Ihr Programm muss deshalb (wenn gefordert) in der Lage sein, eine als Parameter übergebene Datei lesen und ggf. schreiben zu können.

→ Details zu Dateioperationen siehe entsprechende A&P - Vorlesung

Parameter können einem Programm sehr einfach via Kommandozeile übergeben werden. So bewirkt z.B. folgendes Kommandozeilenstatement.

```
./test 123 "Hallo Welt"
```

den Aufruf des Programmes `test` mit den (Zeichenketten-)Parametern `"123"` und `"Hallo Welt"`

Parameterauswertung in C++

```
int main(int argc, const char * argv[]) {  
    /* argc = Anzahl der Argumente des Kommandozeilen-(Shell-)Aufrufs  
       argv[] = Feld von ZEIGERN auf Zeichenketten, die die Parameter  
       der Kommandozeile enthalten  
       argc ist größer >= 1  
       argv[0] ist immer der Name bzw. Pfad des aufgerufenen Programms  
    */  
  
    printf("Ausgabe der Argumente der Kommandozeile\n");  
  
    for (int i = 0; i < argc; i++)  
        printf("%s\n", argv[i]);  
  
    ...  
}
```