



## Aufgabenblatt 2

letzte Aktualisierung: 23. Oktober, 08:49 Uhr

(47167f2625ea1af4e9252f19263fdd337daaa0af)

Ausgabe: **Mittwoch, 18.10.2017**

Abgabe: **Freitag, 20.10.2017, 21:59**

### Thema: Schleifen / Kontrollstrukturen

## Abgabemodalitäten

1. Die Aufgaben des C-Kurses bauen aufeinander auf. Versuche daher bitte Deine Lösung noch am gleichen Tag zu bearbeiten und abzugeben.
2. Alle abzugebenden Quelltexte müssen ohne Warnungen und Fehler auf den Rechnern des tubIT/eecsIT mittels `gcc -std=c99 -Wall` kompilieren.
3. Die Abgabe erfolgt ausschließlich über unser SVN im Abgaben-Ordner. Nur wenn ein Test in Osiris angezeigt wird ist sichergestellt, dass die Abgabe erfolgt ist.
4. Du kannst bis zur Abgabefrist beliebig oft neue Versionen abgeben.
5. Die Abgabe erfolgt in folgendem Unterordner:  
`ckurs-wise1718/Studierende/<L>/<tubIT-Login>/Abgaben/Blatt0<X>`  
wobei `<L>` durch den ersten Buchstabe des TUBIT-Logins und `<X>` durch die Nummer des Aufgabenblattes zu ersetzen sind. Die Ordner werden automatisch angelegt sobald die Abgabe freigeschaltet wird.
6. Benutze für alle Abgaben soweit nicht anders angegeben das folgende Namensschema: `ckurs_blatt0<X>_aufgabe0<Y>.c` wobei `<X>` und `<Y>` entsprechend zu ersetzen sind. Gebe für jede Unteraufgabe genau eine Quellcodedatei ab.
7. Du darfst den Abgabeordner für das Blatt nicht selbst erstellen, das machen wir jeden Morgen kurz nach 8 Uhr!
8. Du musst aber den Befehl `svn up` auf der obersten Verzeichnisebene des Repositories (also in `ckurs-wise1718`) ausführen um alle Änderungen vom Server abzuholen.
9. Im Abgaben-Ordner gelten einige restriktive Regeln. Dort ist nur das Einchecken von Dateien mit den in der Aufgabe vorgegebenen Namen erlaubt, ausserdem werden die Abgabefristen vom Server überwacht. Beachte eventuelle Fehlermeldungen beim SVN-Commit. Lade nur Dateien hoch, die Du selbst bearbeiten sollst, insbesondere also keine Vorgaben.
10. Es gibt einen Ordner 'Arbeitsverzeichnis', in dem Du Dateien für Dich ablegen kannst.
11. Die Ergebnisse der automatischen Tests kannst Du auf OSIRIS einsehen:  
<https://teaching.inet.tu-berlin.de/services/osiris-wise1718/>

## 1. Aufgabe: Ist es eine Primzahl? (2 Punkte)

Schreibe ein Programm, das für eine beliebige Zahl testet ob es eine Primzahl ist. Nutze dafür eine Schleife und gebe mithilfe von `printf` ein "Ja" aus, wenn es sich um eine Primzahl handelt, und "Nein", wenn nicht. Ein beispielhafter Aufruf für die Zahl 105 wird in Listing 1 angezeigt.

### Listing 1: Programmbeispiel

```
> gcc -std=c99 -Wall ckurs_blatt02_aufgabe01.c -o ckurs_blatt02_aufgabe01
> ./ckurs_blatt02_aufgabe01
Ist 105 eine Primzahl?
Nein
```

Zur Erinnerung, eine Primzahl ist eine Zahl, die nur durch sich selbst und durch 1 teilbar ist. Die ersten Primzahlen sind: 2, 3, 5, 7, 11, 13.

Zum Testen ob eine Zahl durch eine zweite Zahl teilbar ist, bietet sich die modulo Operation an. Diese Operation ähnelt einer Division mit Rest. Ein Beispiel für eine Division mit Rest wäre:  $5/2 = 2$  mit Rest 1. Jedoch gibt die Modulo Operation im Unterschied zur Division mit Rest nur den Restbetrag an. Also gilt  $5 \bmod 2 = 1$ . In C wird diese Operation mit einem Prozentzeichen dargestellt: `int ergebnis = 5 % 2;`

Um die Hausaufgabe zu vereinfachen bitten wir Dich die vorgegebene Programmstruktur zu verwenden (siehe Listing 2). Du findest sie auch im SVN unter Aufgaben und im ISIS. Die Abgabe muss folgenden Kriterien entsprechen:

- Die zu testende Zahl wird in der Variable `int nummer` gespeichert.
- Auf dem Terminal werden zwei Zeilen ausgegeben.
- Die zweite Zeile enthält ausschließlich Ja, wenn `nummer` eine Primzahl enthält und ausschließlich Nein wenn das nicht der Fall ist.
- Es dürfen keine weiteren Leerzeilen ausgegeben werden.

Wichtiger Hinweis: Vor dem Einchecken in das SVN ist im obersten Ordner der Arbeitskopie, also dem Verzeichnis `ckurs-wise1718`, der Befehl `svn up` auszuführen, um alle Änderungen im Repository vom Server abzuholen. Danach ist auch das Übungsblatt als pdf-Datei im Materialordner verfügbar und der neue Abgabeordner für das Blatt wird angelegt.

Checke die Abgabe dann im SVN ein, wie unter "Abgabemodalitäten" beschrieben.

### Listing 2: Mögliche Programmstruktur

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     int nummer = 105; // Probiere verschiedene Werte aus: 101, 103, ...
6     printf("Ist_%d_eine_Primzahl?\n", nummer);
7
8     // HIER Code einfügen
9
10    return 0;
11 }
```