

**Imperative Programmierung**  
**Aufgabenblatt A02**  
**Ausgabe: 11. November 2019**  
**Abgabe bis: 24. November 2019, 23:59**

---

**Hinweise:** Die Lösungen der Aufgaben sind als *PDF-Dokument* bzw. *C-Quelltext* mit Hilfe des Versionsverwaltungssystems Subversion (SVN) abzugeben. Platzieren Sie ein PDF-Dokument mit Ihren Antworten im Ordner A02 innerhalb Ihres Gruppenverzeichnisses<sup>1</sup>. Platzieren Sie die C-Quelltexte im Unterordner A02/src. Schreiben Sie in die abgegebenen Dateien Ihren Namen und Ihre Matrikelnummer. Die Referenzsysteme für die Kompilierung sowie Ausführung der Programme sind der PC-Pool und der ITMZ Remote Server (UniComp). Ihre Quelltexte müssen mit den Compilerflags „-ansi -Wall -pedantic“ ohne Warnungen und Fehler übersetzbar sein.

**Verspätete Abgaben oder Abgaben ohne Matrikelnummer werden nicht gewertet!**

---

1. Erläutern Sie die Regeln für erlaubte Variablennamen in C. Der Name `foobar` ist erlaubt, der Name `foo&bar` dagegen nicht – warum?

3 Punkte

2. Könnte es sein, dass sehr lange Variablennamen in C aus bestimmten Gründen problematisch sind? Was sagt der ANSI-Standard bzw. das Kernighan-Ritchie Buch hierzu?

3 Punkte

3. Schreiben Sie ein Programm, das Beträge von Euro in US Dollar umrechnet. Der aktuelle Währungskurs ist:

- 1 Euro entspricht 1.10 USD

Überlegen Sie sich, mit welcher Formel Sie einen Betrag von Euro in USD umrechnen können. Ihr Programm soll so lange Beträge einlesen, bis der Benutzer eine 0 eingegeben hat – dann soll das Programm beendet werden. Wenn der Benutzer einen anderen Wert eingibt, dann soll die Umrechnung in USD und die Ausgabe des Wertes erfolgen.

Ein exemplarischer Programmablauf ist:

```
Bitte geben Sie einen Betrag in Euro ein (Abbruch mit 0): 10.00
Der Betrag entspricht 11.70 USD
Bitte geben Sie einen Betrag in Euro ein (Abbruch mit 0): 20.00
Der Betrag entspricht 23.40 USD
Bitte geben Sie einen Betrag in Euro ein (Abbruch mit 0): -100000.00
Der Betrag entspricht -117000.00
Bitte geben Sie einen Betrag in Euro ein (Abbruch mit 0): 0
```

6 Punkte

---

<sup>1</sup>Ihr Gruppenverzeichnis finden Sie unter  
<https://svn.informatik.uni-rostock.de/lehre/ip2019/groups/Ihr-Gruppenkürzel>.

4. Schreiben Sie ein Programm, das eine positive ganze Zahl in das binäre Zahlensystem übersetzt. Dabei dürfen Sie die Binärstellen in umgedrehter Reihenfolge ausgeben (d.h., niederwertigste Stelle zuerst), so dass Sie etwa für die Zahl  $13_{10}$  die Ausgabe  $1011_2$  erhalten (in normaler Reihung wäre die Ausgabe  $1101_2$ ).

10 Punkte

5. Betrachten Sie das folgende Programm:

```
#include <stdio.h>

main() {
    int w, x, y, z;
    printf("W? ");
    scanf("%d",&w);
    printf("X? ");
    scanf("%d",&x);
    y=1;
    while(y<=x) y=y*w;
    y=y/w;
    while(y > 0) {
        z = x/y;
        printf("%d",z);
        x = x - y*z;
        y = y/w;
    }
    printf("\n");
}
```

- Erläutern Sie, *was* das Programm macht.

7 Punkte

- Erläutern Sie, *wie* es das Programm macht – d. h., welchen Algorithmus das Programm verwendet, um seine Aufgabe zu erfüllen.

8 Punkte

6. In Aufgabe 6 von Aufgabenblatt A01 sollten Sie ein Struktogramm für einen Algorithmus zur Berechnung der Sinusfunktion auf Basis einer Reihenentwicklung angeben. Nun ist Ihre Aufgabe, diesen Algorithmus zu implementieren. Schreiben Sie ein Programm, das einen Winkelwert von der Standardeingabe in eine `double`-Variable einliest, den zugehörigen Sinuswert als `double`-Zahl mit Hilfe Ihres Algorithmus berechnet, und anschließend ausgibt.

Anmerkungen:

- Die Reihenentwicklung erwartet, dass der Winkel in Bogenmaß angegeben wird.
- Für das Einlesen eines Wertes in eine `double`-Variable `x` können sie den Aufruf `scanf("%lf",&x);` verwenden.
- Wenn Sie ihren Algorithmus geschickt formulieren, benötigen Sie für die Reihenentwicklung lediglich eine einzige Schleife.

10 Punkte