



Übungsblatt 2

Abgabe: 08.11.2017, 12:00 Uhr (Postkasten der Veranstaltung und E-Mail an Tutor)

- Dieses Übungsblatt muss im Team abgegeben werden (Einzelabgaben sind nicht erlaubt).
- Bitte zur Angabe von Namen, Übungsgruppe und Teamnummer das **Deckblatt** verwenden!
- Die **Zeitangaben** geben zur Orientierung an, wie viel Zeit für eine Aufgabe später in der Klausur vorgesehen wäre; gehen Sie davon aus, dass Sie zum jetzigen Zeitpunkt wesentlich länger brauchen und die angegebene Zeit erst nach ausreichender Übung erreichen.

* leichte Aufgabe / ** mittelschwere Aufgabe / *** schwere Aufgabe

Aufgabe 5 * (*Wissensfragen, jede Frage 1 Minute*)

Beantworten Sie möglichst knapp und genau in einem Satz die folgenden Fragen.

a) (Ausdrücke - siehe Kapitel "Erste Schritte in C")

1. Ist die Zeichenfolge (10) ein gültiger Ausdruck in C? Wenn nein, warum?
2. Ist die Zeichenfolge 5 - ein gültiger Ausdruck in C? Wenn nein, warum?
3. Ist die Zeichenfolge (5 -) ein gültiger Ausdruck in C? Wenn nein, warum?
4. Ist die Zeichenfolge (5 * 0.01 ein gültiger Ausdruck in C? Wenn nein, warum?
5. Ist die Zeichenfolge 2x ein gültiger Ausdruck in C? Wenn nein, warum?
6. Ist die Zeichenfolge 5 - x ein gültiger Ausdruck in C? Wenn nein, warum?
7. Ist die Zeichenfolge y * x * 5 ein gültiger Ausdruck in C? Wenn nein, warum?
8. Ist die Zeichenfolge 50.0 % 2.00 ein gültiger Ausdruck in C? Wenn nein, warum?
9. Welchen Wert hat der Ausdruck 5 / 4?
10. Welchen Wert hat der Ausdruck - 7.0 / 3.0 ?
11. Welchen Wert hat der Ausdruck 'c' - 'C'?
12. Welchen Wert hat der Ausdruck tolower('X') - 'X'?

-
13. Welchen Wert hat der Ausdruck `'7' - '0'`?
 14. Welchen Wert hat der Ausdruck `'2' - '0'`?
 15. Welchen Wert hat der Ausdruck `printf("Hello World!")`?
 16. Welchen Wert hat der Ausdruck `printf("\\")`?
 17. Welchen Wert hat der Ausdruck `printf("%4i", 0)`?
 18. Welchen Wert hat der Ausdruck `printf("%3i", 10000)`?
 19. Welchen Wert hat der Ausdruck `scanf("%f", &x)`, falls der Benutzer 2.0 eingibt?
 20. Welchen Wert hat der Ausdruck `scanf("Datum: %i.%i.%i", &x, &y, &z)`, falls der Benutzer Datum: 1.1.2016 eingibt?

Aufgabe 6 * (*Bibliotheksfunktionen benutzen - siehe Kapitel "Erste Schritte in C", 16 Minuten*)

Allgemeine Hinweise:

- Verwenden Sie für Benutzereingaben die Funktion `scanf`.
- Sie dürfen bei allen Teilaufgaben von sinnvollen Benutzereingaben ausgehen.
- Mit der Ausgabe eines Programms ist die Ausgabe auf Kommandozeile mit der Funktion `printf` gemeint.
- Mit der Rückgabe einer Funktion ist deren Rückgabewert durch die `return`-Anweisung gemeint.
- Für die Lösung der Aufgaben sind keine `if`- oder `while`-Anweisungen notwendig (diese werden erst in späteren Kapiteln besprochen).
- Die jeweils geeigneten Bibliotheksfunktionen sind eigenständig zu recherchieren. Sehen Sie dazu in den Bibliotheken `math.h` und `ctype.h` nach.

a) (*, 4 Minuten)

Schreiben Sie ein C-Programm, das vom Benutzer zwei ASCII-Zeichen einliest. Das Programm soll mit Hilfe einer geeigneten Bibliotheksfunktion überprüfen, ob beide eingegebene Zeichen eine Ziffer waren. Im positiven Fall soll es den Wert 1 ausgeben, im negativen Fall, falls eines der Zeichen keine Ziffer war, den Wert 0.

b) (*, 4 Minuten)

Schreiben Sie ein C-Programm, das vom Benutzer ein ASCII-Zeichen einliest. Falls das Zeichen ein Großbuchstabe ist, soll das Programm das Zeichen mit Hilfe einer geeigneten Bibliotheksfunktion in einen Kleinbuchstaben umwandeln und ausgeben, und ansonsten unverändert lassen und ausgeben.

c) (*, 4 Minuten)

Schreiben Sie ein C-Programm, das vom Benutzer einen Dezimalbruch einliest. Das Programm soll mit Hilfe einer geeigneten Bibliotheksfunktion den kleinsten ganzzahligen Wert bestimmen, der nicht kleiner ist als der eingegebene Dezimalbruch. Es soll diesen ganzzahligen Wert ohne Nachkommastellen ausgeben.

d) (*, 4 Minuten)

Schreiben Sie ein C-Programm, das vom Benutzer einen Dezimalbruch zwischen den Werten -1 und 1 einliest. Das Programm soll mit Hilfe einer geeigneten Bibliotheksfunktion den Arcus Sinus

dieses Werts bestimmen und dann auf 4 Nachkommastellen genau ausgeben.

Aufgabe 7 ** (Algorithmen - siehe Kapitel "Erste Schritte in C", 20 Minuten)

a) (Variablenwerte verfolgen *, 5 Minuten)

Betrachten Sie den folgenden in Pseudocode gegebenen Algorithmus

<p>Eingabe : $x, y \in \mathbb{N}$ $k \leftarrow 0$; solange $x + 1 > y$ tue $x \leftarrow x - y$; $k \leftarrow k + 1$; Ausgabe : k</p>

Verfolgen Sie in einer Tabelle wie auf Folie 52 (Kapitel 2) die Werte von x , y und k bei der Verarbeitung dieses Algorithmus für die Eingabe $x = 20$ und $y = 6$. Welche mathematische Operation berechnet der Algorithmus?

b) (Algorithmus durch C-Funktion realisieren **, 4 Minuten)

Realisieren Sie den Algorithmus aus Aufgabe a) durch eine C-Funktion mit dem Namen `comp` (natürlich ohne direkte Benutzung der berechneten mathematischen Funktion). Benutzen Sie folgende Vorlage, um die Funktion in einem Hauptprogramm zu testen:

```
#include <stdio.h>

/*
 * Hier fügen Sie die Definition der comp-Funktion ein
 */

int main(void)
{
    int x, y;
    printf("Bitte geben Sie zwei positive ganze Zahlen ein\n");
    printf("(getrennt durch ein Leerzeichen):\n");
    scanf("%i %i", &x, &y);
    /*
     * Hier fügen Sie die printf-Ausgabe des Rückgabewerts
     * der comp-Funktion ein
     */
    return 0;
}
```

c) (Algorithmus entwerfen **, 5 Minuten)

Entwerfen Sie einen Algorithmus in der Pseudocode-Notation aus Unterkapitel 3.1, der für eine positive ganze Zahl x größer 2 überprüft, ob diese Zahl eine Primzahl ist. Im positiven Fall soll 1 ausgegeben werden, falls x keine Primzahl ist soll 0 ausgegeben werden.

d) (Algorithmus durch Maschinenprogramm realisieren ***, 6 Minuten)

Nehmen Sie an, Ihnen würden genau die in Kapitel 3.6 definierten Maschinenbefehle zur Verfügung stehen. Realisieren Sie den folgenden Algorithmus (wie in Kapitel 3.8 gezeigt) durch ein Maschinenprogramm unter ausschließlicher Benutzung dieser Befehle.

Eingabe : $n \in \mathbb{N}$
 $e \leftarrow 0$;
solange $n > 0$ **tue**
 $e \leftarrow e + 2 * n$;
 $n \leftarrow n - 1$;
Ausgabe : e

Gehen dazu davon aus, dass der Wert von n in SZ 1000 und der Wert von e in SZ 3000 gespeichert ist, und dass das Maschinenprogramm ab Speicherzelle 2001 im Arbeitsspeicher abgelegt ist.

Aufgabe 8 ** (*Maschinenprogramme und Rechnerarchitektur, 16 Minuten*)

a) (Maschinenprogramm interpretieren **, 4 Minuten)

In der Speicherzelle (SZ) 1001 sei eine positive ganze Zahl gespeichert. Betrachten Sie das folgende im Arbeitsspeicher in den SZ 2000 bis 2005 abgelegte Maschinenprogramm:

```
SZ 2000: INIT 1002
SZ 2001: SPRUNG 1001,2005
SZ 2002: ADD 1002,1001
SZ 2003: DEKREMENT 1001
SZ 2004: SPRUNG 2001
SZ 2005: AUSGABE 1002
```

- Was ist die Ausgabe des Programms, falls bei Programmstart in SZ 1001 der Wert 4 gespeichert ist?
- Welche Berechnung wird durch dieses Programm realisiert?

b) (Maschinenprogramm entwerfen ***, 6 Minuten)

In den Speicherzellen (SZ) 1006 und 1007 seien positive ganze Zahlen gespeichert. Realisieren Sie die Addition der zwei in den SZ 1006 und 1007 gespeicherten positiven ganzen Zahlen durch ein Maschinenprogramm, ohne den ADD-Befehl zu benutzen. Nehmen Sie dabei an, dass das Programm ab SZ 2001 im Arbeitsspeicher abgelegt ist.

c) (Fetch/Decode/Execute-Zyklus *, 6 Minuten)

Betrachten Sie den folgenden in SZ 400 gespeicherten Maschinenbefehl eines Maschinenprogramms:

```
SZ 400: ADD 103,200
```

Nehmen Sie an, der Inhalt des PC wäre die Adresse 400. Beschreiben Sie die folgende Ausführung eines Fetch/Decode/Execute-Zyklus durch die CPU, wobei Sie auf eine detaillierte Darstellung der Kommunikation zwischen Speicherwerk und Steuerwerk verzichten.

Es gibt folgende Arten von **Zusatzaufgaben**:

- Aufgaben zu **Schulstoff**, der Voraussetzung zum Verständnis der Vorlesung ist. Es wird davon ausgegangen, dass Sie diesen (Schul-)Stoff bereits beherrschen.
- **Alte Klausuraufgaben** zu Themen des Übungsblatts.

Besprechungen / Lösungen zu Zusatzaufgaben:

- Alle Zusatzaufgaben sind **unbewertet** und dementsprechend nicht abzugeben.
- Alle Zusatzaufgaben werden in der **Globalübung** besprochen.

Zusatzaufgabe B (*Klausuraufgaben aus dem Wintersemester 2016/2017*)

i) (*Algorithmus durch Maschinenprogramm realisieren*, 6 Punkte)

Realisieren Sie den nachfolgenden Algorithmus durch ein Maschinenprogramm, das ab Adresse 4 gespeichert ist, wobei Sie ausschließlich Maschinenbefehle aus der Vorlesung verwenden.

<pre>Eingabe : $a, b \in \mathbb{N}$ $e \leftarrow 0$; solange $b > 0$ tue $e \leftarrow e + a$; $b \leftarrow b - 1$; Ausgabe : e</pre>

Gehen Sie dazu von folgender Zuordnung von Variablen zu Speicheradressen aus:

- a : Adresse 1
- b : Adresse 2
- e : Adresse 3

ii) (*Fetch/Decode/Execute-Zyklus*, 5 Punkte)

Betrachten Sie den folgenden an Adresse 2 gespeicherten Maschinenbefehl eines Maschinenprogramms:

2: INKREMENT 1

Nehmen Sie an, der Inhalt des PC wäre die Adresse 2. Beschreiben Sie die Ausführung eines *Fetch/Decode/Execute* - Zyklus durch die CPU zur Abarbeitung des obigen Maschinenbefehls, wobei Sie auf eine detaillierte Darstellung der Kommunikation zwischen Speicherwerk und Steuerwerk *verzichten*.