



Übungsblatt 2

Allgemeine Hinweise:

- Achten Sie bei allen Programmieraufgaben auf *Kompilierbarkeit* und *Einhaltung der Coding Conventions* (werden zu jedem Thema in den Vorlesungsfolien erklärt); auch dann, wenn es nicht explizit im Aufgabentext gefordert ist.
- Gehen Sie davon aus, dass alle Programme in der Programmiersprache C zu erstellen sind.
- Kompilieren Sie alle Ihre Programme mit den folgenden *Compiler-Schaltern*:
-Wall -Wextra -ansi -pedantic
Achten Sie darauf, dass trotz Verwendung dieser Schalter keine Fehler-/Warnmeldungen erzeugt werden.

In den folgende Programmieraufgaben werden Sie sich mit dem Datentyp `int`, arithmetischen Rechenoperatoren und Möglichkeiten der formatierten Ausgabe mit `printf` beschäftigen.

Zur Eigenrecherche können u.a. die folgenden zwei Internetseiten dienen:

- http://de.wikibooks.org/wiki/C-Programmierung:_Einfache_Ein-_und_Ausgabe
- http://de.wikibooks.org/wiki/C-Programmierung:_Ausdr%C3%BCcke_und_Operatoren

In den Aufgaben 1 - 3 soll in jeder Teilaufgabe ein C-Programm mit einer eigenen `main`-Funktion erstellt und dazu die in Worten beschriebenen Anweisungen in C-Anweisungen umformuliert werden. Dabei ist jeder Satz in genau eine C-Anweisung zu überführen. Beispielsweise soll die in Worten beschriebene Anweisung

- Deklarieren Sie drei Variablen `x`, `y` und `z` vom Typ `int`.

in die C-Anweisung

- `int x, y, z;`

überführt werden.

In Aufgabe 4 sollen Fehler in vorgegebenen Programmen gefunden und verbessert werden.

Aufgabe 1 *(Ganze Zahlen)*

a)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie `n` den Wert `9` zu.
- Geben Sie den Wert von `n` mit der Funktion `printf` aus.

b)

- Deklarieren Sie eine `int`-Variable `m`.
- Weisen Sie `m` den Wert `+9` zu.
- Geben Sie den Wert von `m` mit der Funktion `printf` aus.

c)

- Deklarieren Sie eine `int`-Variable `k`.
- Weisen Sie `k` den Wert `-9` zu.
- Geben Sie den Wert von `k` mit der Funktion `printf` aus.

d)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie `n` den Wert `0` zu.
- Geben Sie den Wert von `n` mit der Funktion `printf` aus.

e)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie `n` den Wert `-0` zu.
- Geben Sie den Wert von `n` mit der Funktion `printf` aus.

f)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie `n` den Wert `INT_MAX` zu.
- Geben Sie den Wert von `n` mit der Funktion `printf` aus.

g)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie `n` den Wert `-INT_MAX` zu.
- Geben Sie den Wert von `n` mit der Funktion `printf` aus.

h)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie `n` den Wert `INT_MIN` zu.
- Geben Sie den Wert von `n` mit der Funktion `printf` aus.

i)

-
- Deklarieren Sie eine `int`-Variable `n`.
 - Erzeugen Sie mit der Funktion `rand` eine ganze Zufallszahl und weisen Sie `n` den Wert der Zufallszahl zu.
 - Geben Sie den Wert von `n` mit der Funktion `printf` aus.

Aufgabe 2 (*Arithmetische Rechenausdrücke*)

a)

- Deklarieren Sie drei `int`-Variablen `k`, `m` und `n`.
- Weisen Sie `k` den Wert 1 zu.
- Weisen Sie `m` den Wert 2 zu.
- Weisen Sie `n` den Wert der Addition von `k` und `m` zu.
- Geben Sie mit der Funktion `printf` die Werte von `k`, `m` und `n` in der Form `k + m = n` aus.

b)

- Deklarieren Sie drei `int`-Variablen `k`, `m` und `n`.
- Weisen Sie `k` den Wert 1 zu.
- Weisen Sie `m` den Wert 2 zu.
- Weisen Sie `n` den Wert der Subtraktion von `k` und `m` zu.
- Geben Sie mit der Funktion `printf` die Werte von `k`, `m` und `n` in der Form `k - m = n` aus.

c)

- Deklarieren Sie drei `int`-Variablen `k`, `m` und `n`.
- Weisen Sie `k` den Wert 1 zu.
- Weisen Sie `m` den Wert 2 zu.
- Weisen Sie `n` den Wert der Multiplikation von `k` und `m` zu.
- Geben Sie mit der Funktion `printf` die Werte von `k`, `m` und `n` in der Form `k * m = n` aus.

d)

- Deklarieren Sie drei `int`-Variablen `k`, `m` und `n`.
- Weisen Sie `k` den Wert 1 zu.
- Weisen Sie `m` den Wert 2 zu.
- Weisen Sie `n` den Wert der ganzzahligen Division von `k` und `m` zu.
- Geben Sie mit der Funktion `printf` die Werte von `k`, `m` und `n` in der Form `k / m = n` aus.

e)

- Deklarieren Sie drei `int`-Variablen `k`, `m` und `n`.
- Weisen Sie `k` den Wert 1 zu.
- Weisen Sie `m` den Wert 2 zu.

-
- Weisen Sie `n` den Wert des Rests bei ganzzahliger Division von `k` und `m` zu.
 - Geben Sie mit der Funktion `printf` die Werte von `k`, `m` und `n` in der Form `k % m = n` aus.

f)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie `n` den Wert 10 zu.
- Erhöhen Sie dann den Wert von `n` um 1.
- Geben Sie den Wert von `n` aus.

g)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie `n` den Wert 10 zu.
- Verdoppeln Sie dann den Wert von `n`.
- Geben Sie den Wert von `n` aus.

h)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie `n` den Wert 10 zu.
- Halbieren Sie dann den Wert von `n`.
- Geben Sie den Wert von `n` aus.

i)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie `n` den Wert 10 zu.
- Verringern Sie dann den Wert von `n` um 1.
- Geben Sie den Wert von `n` aus.

j)

- Deklarieren Sie drei `int`-Variablen `k`, `m` und `n`.
- Weisen Sie `k` und `m` jeweils eine Zufallszahl zu, die sie mit der Funktion `rand` erzeugen.
- Weisen Sie `n` den Wert der Addition von `k` und `m` zu.
- Geben Sie mit der Funktion `printf` die Werte von `k`, `m` und `n` in der Form `k + m = n` aus.

k)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie `n` eine Zufallszahl zu, die sie mit der Funktion `rand` erzeugen.
- Halbieren Sie dann den Wert von `n`.
- Geben Sie den Wert von `n` aus.

Aufgabe 3 *(Auswertungsreihenfolge und Overflow)*

a)

- Deklarieren Sie drei `int`-Variablen `k`, `m` und `n` und weisen Sie `k` den Wert 2, `m` den Wert 3 und `n` den Wert 5 zu.
- Geben Sie den Wert der folgenden Rechnung aus: Multiplizieren Sie zuerst `k` und `m`, und addieren Sie dann das Ergebnis zu `n`.

b)

- Deklarieren Sie drei `int`-Variablen `k`, `m` und `n` und weisen Sie `k` den Wert 2, `m` den Wert 3 und `n` den Wert 5 zu.
- Geben Sie den Wert der folgenden Rechnung aus: Addieren Sie zuerst `m` und `n`, und multiplizieren Sie dann das Ergebnis mit `k`.

Die Rechnungen in den Teilaufgaben a) und b) unterscheiden sich in der Auswertungsreihenfolge der Rechenoperationen und führen zu unterschiedlichen Ergebnissen.

c)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie ihr den Wert `INT_MAX` zu.
- Erhöhen Sie dann den Wert von `n` um 1.
- Geben Sie den Wert von `n` aus.

d)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie ihr den Wert `INT_MAX` zu.
- Erhöhen Sie dann den Wert von `n` um 2.
- Geben Sie den Wert von `n` aus.

e)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie ihr den Wert `INT_MAX` zu.
- Erhöhen Sie dann den Wert von `n` um `INT_MAX`.
- Geben Sie den Wert von `n` aus.

f)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie ihr den Wert `INT_MIN` zu.
- Verringern Sie dann den Wert von `n` um 1.
- Geben Sie den Wert von `n` aus.

Die Rechnungen in den Teilaufgaben c) - f) führen zu einem sog. **Overflow**. Das bedeutet, dass der Wertebereich des benutzten Datentyps **int** verlassen wird. Das führt zwar zu keinem Fehler beim Kompilieren oder im Programmablauf (für den Compiler ist das Programm korrekt und es gibt auch keine Programmabbruch zur Laufzeit bei Ausführung des Programms), aber es kommt zu unerwarteten Rechenergebnissen.

Aufgabe 4 (Syntaxfehler)

Die folgenden C-Programme sollten direkt nach dem Kompilieren auf der Kommandozeile ausführbar sein. Leider haben sich einige Fehler eingeschlichen! Untersuchen Sie die zugehörige(n) Fehlermeldung(en) des Compilers, schreiben Sie jeweils eine fehlerbereinigte Version des Programms und fügen Sie einen Kommentar mit einer Beschreibung des Fehlers ein.

a)

```
1 int main(void)
2 {
3     return 0
4 }
```

b)

```
1 int main(void)
2 {
3     Return 0;
4 }
```

c)

```
1 int Main(void)
2 {
3     return 0;
4 }
```

d)

```
1 int main(void)
2 {
3     return 0;
```

e)

```
1 int main(void)
2 {
3     printf("Hallo");
4     return 0;
5 }
```

f)

```
1 include <stdio.h>
3 int main(void)
4 {
5     printf("Hallo");
6     return 0;
7 }
```

g)

```
1 #Include <stdio.h>
3 int main(void)
4 {
5     printf("Hallo");
6     return 0;
7 }
```

h)

```
1  #include <stdio.h>
3  int main(void)
4  {
5      Printf("Hallo");
6      return 0;
7  }
```

i)

```
1  #include <stdio.h>;
3  int main(void)
4  {
5      printf("Hallo");
6      return 0;
7  }
```

j)

```
1  #include <stdio.h>
3  int main(void)
4  {
5      printf("%i");
6      return 0;
7  }
```

k)

```
1  #include <stdio.h>
3  int main(void)
4  {
5      printf(Hallo);
6      return 0;
7  }
```

l)

```
1  #include <stdio.h>
3  int main(void)
4  {
5      printf("Hallo", 5);
6      return 0;
7  }
```

m)

```
1  #include <stdio.h>
3  int main(void)
4  {
5      printf("%i", x);
6      return 0;
7  }
```

n)

```
1  #include <stdio.h>
3  int main(void)
4  {
5      x = 5;
6      printf("%i", x);
7      return 0;
8  }
```

o)

```
1  #include <stdio.h>
3  int main(void)
4  {
5      Int x = 5;
6      printf("%i", x);
7      return 0;
8  }
```

p)

```
1  #include <stdio.h>
3  int main(void)
4  {
5      int x = 5;
6      printf("%i", 2x);
7      return 0;
8  }
```

q)

```
1  #include <stdio.h>
3  int main(void)
4  {
5      printf("%i", ((2 + 3) * 5 + (1 - 7)));
6      return 0;
7  }
```

r)

```
1  #include <stdio.h>
3  int main(void)
4  {
5      printf("%i", 8 : 4);
6      return 0;
7  }
```