

EINFÜHRUNG IN DIE PROGRAMMIERUNG
MIT DER PROGRAMMIERSPRACHE C
(**EifP**, WS2025/26)

Prof. Dr. Thomas Gabel
Roman Ahmad, Emre Özöner, Gina Romanazzi

Aufgabenblatt 3

Aufgabe 9: Algorithmen

Inhalte und Sprachkonzepte der vorliegenden Aufgabe

- Eigenschaften von Algorithmen

Aufgabenstellung

Klassifizieren Sie die folgenden Algorithmen, die alle keine Benutzereingaben erwarten, indem Sie folgende Fragen beantworten:

- a) Ist der Algorithmus terminierend / nicht terminierend?
- b) Ist der Algorithmus deterministisch / nicht deterministisch?
- c) Ist der Algorithmus determiniert / nicht determiniert?
- d) Ist der Algorithmus iterativ / nicht iterativ?

Algorithmus 1

1. Wähle eine zufällige natürliche Zahl z größer null.
2. Multipliziere diese Zahl z mit 5 und speichere das Ergebnis in x .
3. Addiere die gewählte Zahl z zu dem Produkt x und speichere das Ergebnis in y .
4. Dividiere das Ergebnis y durch die ursprünglich gewählte Zahl z .
5. Gib das Ergebnis y aus.

Algorithmus 2

1. Wähle einen zufälligen Grenzwert g aus dem Intervall $[0,1]$.
2. Setze den Wert von n auf 2.

3. Wenn n den Grenzwert g unterschreitet, gehe nach 6.
4. Teile n durch 2 und speichere das Ergebnis in n .
5. Gehe zurück nach 3.
6. Gebe den Wert von n aus.

Algorithmus 3

1. Setze n auf 1.
2. Wenn n größer als 5 ist, gehe nach 6.
3. Multipliziere n mit 2 und speichere das Ergebnis in n .
4. Wenn n^2 größer ist als 15, setze n auf 1.
5. Gehe zurück nach 2.
6. Gib den Wert von n aus.

Aufgabe 10: Variationen von “Hello World!”

Inhalte und Sprachkonzepte der vorliegenden Aufgabe

- Hello World!
- Eingabe und Ausgabe

Aufgabenstellung

a) **Programm hello1**

Schreiben Sie ein Programm mit dem Name `hello1`, das die Zeichenkette „Hello, World!“ auf dem Terminal ausgibt. Benutzen Sie dabei die `printf`-Funktion. Stellen Sie sicher, dass am Ende der Ausgabe ein Zeilenumbruch erfolgt. Übersetzen Sie Ihr Programm wie folgt:

```
gcc hello1.c -o hello1
```

b) **Programm hello2**

Schreiben Sie ein Programm, das Folgendes auf dem Bildschirm ausgibt:

```
/* Das ultimative Hello-World-Programm */
#include <stdio.h>
int main(void)
{
    printf("Hallo Welt!\n");
}
```

Das bedeutet, das Programm `hello2` soll in der Lage sein, das hier angegebene, 6-zeilige Hallo-Welt-Programm auf den Bildschirm auszugeben.

c) **Programm hello3**

Modifizieren Sie `hello1.c` derart, dass zunächst die Ausgabe „Enter your first name:“ erscheint, danach von der Standardeingabe (`stdin`) mit der Funktion `scanf` ein Name eingelesen wird und schließlich das Programm mit der Ausgabe „Hello, <your name>!“ beendet wird. Für den Platzhalter <your name> sollte der eingegebene Name ausgegeben werden. Der Name soll auf ein Wort beschränkt sein.

Bemerkung: Die Funktion `scanf` sollte wie folgt aufgerufen werden:

```
char s[100]; /* N.B. strings are arrays of chars in C */
/* ... maybe some intervening code ... */
scanf("%99s", s);
```

Aufgabe 11: Datentypen

Inhalte und Sprachkonzepte der vorliegenden Aufgabe

- Speicherbedarf von Datentypen

Aufgabenstellung

Schreiben Sie ein Programm, das mit Hilfe des `sizeof`-Operators für die Datentypen `char`, `int`, `short int`, `float`, und `double` ausgibt, wie viel Speicher eine Variable des jeweiligen Typs belegt. Stellen Sie sicher, dass für jeden Datentyp der erforderliche Speicher in *Bit* (nicht Byte) ausgegeben wird.

Zusatzaufgabe: Erweitern Sie das Programm nun so, dass die obere Schranke des Wertebereiches von `unsigned short int` und `unsigned char` ausgegeben wird.

Aufgabe 12: Rechenaufgaben

Inhalte und Sprachkonzepte der vorliegenden Aufgabe

- arithmetische Ausdrücke

Aufgabenstellung

Schreiben Sie ein C-Programm, das die folgenden mathematischen Ausdrücke berechnet und ausgibt (je ein Ergebnis pro Zeile). Dabei sollen nur die vier Grundrechenarten zur Anwendung kommen. Verwenden Sie dazu folgende Variablendefinitionen und Initialisierungen:

```
double a=2.0, b=3.0, c=0.4E1, p=3.14;
```

a) $5 \cdot a + 76 + 3.1$

b) $\frac{a+6}{2 \cdot p \cdot a + \frac{1}{a}}$

c) $a + \frac{b - \frac{2}{c}}{a + \frac{1}{a - \frac{1}{b}}}$

d) $(-a)^2$

e) a^{-2}

Beispiel: Für den mathematischen Ausdruck $3 \cdot a + b \cdot c$ könnte Ihre Lösung so aussehen:

```
#include <stdio.h>
int main(void)
{
    double a=2.0, b=3.0, c=0.4E1, p=3.14;
    double ergebnis;
    ergebnis = 3*a + b*c;
    printf("Es ist 3*a+b*c = 3*%lf+%lf*%lf = %lf\n", a, b, c, ergebnis);
    return 0;
}
```

Aufgabe 13: Grundrechenarten

Inhalte und Sprachkonzepte der vorliegenden Aufgabe

- Einlesen mit `scanf`
- einfache arithmetische Operatoren
- Datentypen und ihre Bedeutung

Aufgabenstellung

- a) Schreiben Sie ein Programm, dass zwei `int`-Zahlen `a` und `b` einliest und die Operationen `a+b`, `a-b` sowie `a*b` durchführt. Die Ausgabe soll folgende Form haben (am Beispiel für `a=2` und `b=5`):

```
2 + 5 = 7
2 - 5 = -3
2 * 5 = 10
```

- b) Ändern Sie anschließend den Datentyp von `a` und `b` in `float` um und führen Sie dieselben Operationen durch.
- c) Behalten Sie nun den Datentyp `float` aus Teilaufgabe b) bei und erweitern Sie Ihr Programm um eine Division. Zusätzlich soll also im obigen Beispiel eine Zeile mit dem Divisionsergebnis (0.4) ausgegeben werden.