

Programmierung, Algorithmen, Datenstrukturen 1

a.o. Univ.-Prof. Dr. Martin Welk
Dipl.-Ing. Elias Tappeiner
Institut für Biomedizinische Bildanalyse

UNIT
WS 2021/22
Mech-BSc/ET-BSc

Hausübung 2

Abgabe: 8. November 2021, 23.55 Uhr in Moodle

Es wird empfohlen, die Übungen in Gruppen (max. 3 Studierende) zu bearbeiten und gemeinsam abzugeben.

Hinweise zur Abgabe

- Gruppenarbeiten werden von **einer/m** Studierenden in Moodle eingereicht; die Namen **aller** Beteiligten sind bei der Abgabe zu vermerken! (Empfehlung: Kommentar im Quelltextfile)
- Laden Sie in Moodle die C-Quellcodedatei hoch, bitte keine Codeblocks-Projektdaten o. Ä.

Hinweise zur Bewertung

In die Bewertung der Bearbeitungen fließen ein

- Syntaktische Korrektheit: Das Programm muss fehlerfrei übersetzbar und ausführbar sein.
- Funktionalität: Das Programm muss den vorgegebenen Algorithmus/die vorgegebene Berechnung korrekt ausführen.
- Programmstruktur: Das Programm muss sinnvoll gegliedert (Funktionen!), übersichtlich formatiert (Einrückungen!) und verständlich kommentiert sein.

→

Aufgabe 2.1

Schreiben Sie ein C-Programm, das vier ganze Zahlen $a \geq 0$, $b \geq 0$, $c \geq 0$, $d > 0$ als Konsoleneingaben abfragt und dann diejenigen ganzen Zahlen n mit $0 \leq n < d$ bestimmt, für die

$$an^2 + bn + c \text{ ohne Rest durch } d \text{ teilbar}$$

ist. (Die Berechnung soll erfolgen, indem alle Werte von n in einer Schleife nacheinander durchprobiert werden und diejenigen, für die die Teilbarkeit zutrifft, ausgegeben werden.)

Beispiele:

$$a = 1, b = 0, c = 7, d = 8 \rightarrow n = 1, 3, 5, 7.$$

$$a = 1, b = 5, c = 3, d = 19 \rightarrow \text{keine Lösungen für } n.$$

$$a = 2, b = 5, c = 3, d = 19 \rightarrow n = 8, 18.$$

Achten Sie auf das Abfangen unzulässiger Eingaben (negative Zahlen oder $d = 0$) sowie auf eine sinnvolle Funktionsgliederung. Wenn die gesamte Eingabe und Berechnung in `main()` erfolgt, kann keine volle Punktzahl erreicht werden.

Aufgabe 2.2

Schreiben Sie ein C-Programm, das die Koordinaten zweier Vektoren $u, v \in \mathbb{R}^3$ als Konsoleneingaben entgegennimmt und anschließend das Skalarprodukt $u \cdot v$, das Vektorprodukt $u \times v$ sowie den Winkel zwischen den Vektoren u und v in Grad bestimmt.

Beispiel eines möglichen Programmlaufs:

```
./vektorberechnung
Ersten Vektor eingeben
Erste Koordinate: 1.3
Zweite Koordinate: 4
Dritte Koordinate: -2.1
Zweiten Vektor eingeben
Erste Koordinate: 3.2
Zweite Koordinate: 0
Dritte Koordinate: 5
Skalarprodukt: -6.340000
Vektorprodukt: (20.000000, -13.220000, -12.800000)
Winkel: 103.131259 Grad
```

Erinnerung: Für Vektoren

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

gilt

$$\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + u_2 v_2 + u_3 v_3,$$

$$\mathbf{u} \times \mathbf{v} = \begin{pmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{pmatrix}$$

sowie

$$\sphericalangle(\mathbf{u}, \mathbf{v}) = \arccos \left(\frac{\mathbf{u} \cdot \mathbf{v}}{\sqrt{\mathbf{u} \cdot \mathbf{u}} \cdot \sqrt{\mathbf{v} \cdot \mathbf{v}}} \right).$$

Die Funktion `arccos` ist in der C-Standardbibliothek (Header `math.h` einbinden!) als `acos` verfügbar und liefert ihr Ergebnis im Bogenmaß (Radiant) zurück; d. h. das Ergebnis muss noch in Grad umgerechnet werden.

Achten Sie auf das Abfangen von Sonderfällen (was geht schief, wenn \mathbf{u} oder \mathbf{v} der Nullvektor ist?) sowie auf eine sinnvolle Funktionsgliederung. Wenn ungültige Werte (NaN – Not a number) ausgegeben werden oder die gesamte Eingabe und Berechnung in `main()` erfolgt, kann keine volle Punktzahl erreicht werden.