Modellierung und Programmierung 2

Hinweise Abgabe: https://moodle2.uni-leipzig.de/course/view.php?id=3907

Übungsserie 1

Programmierung mit C

Hinweis: Bitte beachten Sie eventuell benötigte Bibliotheken zum Linken. Beispielsweise `gcc -std=c11 -o hallo hallo.c -lm` *für das Linken gegen die Mathebibliothek. Nutzen Sie* `-std=c99` *falls* `c11` *nicht verfügbar ist. Jede Aufgabe ist 3 Punkte wert.*

Aufgabe 1 Schreiben Sie ein C-Programm `klausur.c` zur Auswertung von Klausurergebnissen. Das Programm soll aus folgenden drei Funktionen bestehen: Die erste Methode `avg` soll als Rückgabewert (float) das arithmetische Mittel aller Noten errechnen. Die zweite Methode `geom` soll als Rückgabewert (float) das geometrische Mittel aller Noten berechnen. Und die dritte Methode `median` soll als Rückgabewert (float) den Median ($\tilde{x}$) zurückgeben. Jede dieser Funktionen braucht als Parameter das Feld der Noten und die Notenanzahl. Der Median ist dabei das mittlere Element der sortierten Notenfolge:

$$\tilde{x} = median(x_1, ..., x_n) = \begin{cases} x_{\frac{n+1}{2}} & n \text{ ungerade} \\ \frac{1}{2}(x_{\frac{n}{2}} + x_{\frac{n}{2}+1}) & n \text{ gerade.} \end{cases}$$

Testen Sie exemplarisch alle drei Funktionen im Hauptprogramm an folgenden statisch zu allozierenden Beispielen:

- `{1,2,3,4,5,6}`
- `{1,2,3,4,3,2,1,2,3,4,3,2,1,5,6}`

Hinweise: Achten Sie auf Indexverschiebungen, da Felder immer mit Index Null beginnen. Sie dürfen die gesamte *Standard* Biliothek (aber keine weiteren Bibliotheken) verwenden. Für eventuelles Sortieren dürfen Sie die Felder verändern.

Aufgabe 2 Entwerfen Sie ein C-Programm `cramer.c`, welches bei Eingabe der Parameter a, b, c, d, e, f das folgende lineare Gleichungssystem mit zwei Unbekannten (x, y) durch die Cramersche Regel löst:

$$a \cdot x + b \cdot y = e$$
$$c \cdot x + d \cdot y = f$$

Hinweis: Nutzen Sie die explizite Lösungsvorschrift für Gleichungssysteme 2. Ordnung. Testen Sie das Programm mit den Werten $a = 3,\ b = 1.52,\ c = 2,\ d = 1.02,\ e = 1,\ f = 1$ und ersetzen Sie d in einem weiteren Testlauf durch 1.03. Was fällt auf?

Aufgabe 3 Ein Palindrom ist eine Zeichenkette, welche vom Start zum Ende und vom Ende zum Start gelesen gleich ist. Beispiele: `OTTO`, `ANNA` oder auch `12345678987654321`. Groß- und Kleinschreibung sei hierbei unterschiedlich, also `OTTo` wäre somit *kein* Palindrom. Schreiben Sie ein Programm `palindrom.c`, welches eine Zeichenkette überprüft, ob sie ein Palindrom ist. Verwenden Sie dabei folgende Funktionssignatur:

```
bool isPalindrome( const char * str );
```

Geben Sie anschließend und auf der Kommandozeile `true` oder `false` aus. Das Programm soll dabei in $O(n)$ arbeiten, wobei n die Länge der Zeichenkette ist.

Hinweis: In C sind Zeichenketten stets mit `'\0'` zu terminieren. Der Datentyp `bool` ist in `stdbool.h` definiert.

Aufgabe 4 Das "Sieb des Erathostenes" ist ein Verfahren zur Bestimmung aller Primzahlen zwischen 2 und $n \in \mathbb{N}$, welches ohne Division auskommt. Hierzu werden alle ganzen Zahlen von 2 bis n notiert. Wir nennen diese Folge das Sieb. Sei p der kleinste noch nicht gestrichene Wert im Sieb. p ist mit Sicherheit eine Primzahl. Wir streichen im Sieb alle Vielfachen $j \cdot p \leq n$ mit $j \in \mathbb{N} \setminus \{0\}$. Anschließend ist der kleinste nicht gestrichene Wert q die nächste gesuchte Primzahl. Jetzt verfahren wir für q wie zuvor mit p. Dieses Schritte wiederholen wir solange bis wir zu einer Primzahl kommen, für die $q^2 > n$ gilt. Da dann im Sieb keine Produkte $a \cdot b$ mit $a > q$ und $b > q$ stehen können, denn solche Produkte sind wegen $q > \sqrt{n}$ größer als n, und da alle Produkte $a * b$ mit $a \leq q$ und/oder $b \leq q$ bereits gestrichen wurden, enthält das Sieb nur noch die Primzahlen zwischen 2 und n.

Schreiben Sie ein C-Programm `sieb.c`, das den "Sieb des Erathostenes"-Algorithmus implementiert. Verwenden Sie als Datenstruktur für das Sieb ein dynamisch alloziertes eindimensionales Array.
Eingabe: n
Ausgabe: alle Primzahlen zwischen 2 und n

Zusatzaufgabe 1 Gegeben sei eine Zahlenfolge in expliziter Bildungsvorschrift:

$$g(n) = \frac{2}{3} \cdot \left(1 - \left(-\frac{1}{2}\right)^n\right),\ n \in \mathbb{N}_0$$

Notieren Sie sich (mindestens) die ersten fünf Werte der Folge. Konstruieren Sie nur auf Grundlage dieser Werte eine *rekursive* Funktion in dem Programm `folge.c`, welche für alle $n \geq 0$ ein zu $g(n)$ identisches Ergebnis liefert. Verifizieren Sie ferner die Gültigkeit Ihrer rekursiven Funktion, indem Sie einen mathematisch sauberen Beweis führen, dass Ihre rekursive Funktion und $g(n)$ für alle ganzen Zahlen $n \geq 0$ den gleichen Wert liefern.