

Programmierung 1

Übungsblatt Woche 8 - 10. - 16. Dezember 2025

Testat!

1. Maximum (blatt8_1.c)

Berechnen Sie das Maximum aus einer Menge von Ganzzahlen (auch negative), welche in einer selbst erstellten Textdatei (z.B. *werte.txt*) abgespeichert sind.¹

Lesen Sie dazu die Zahlen aus der Datei ein, wobei in letzterer beliebig viele Zahlen stehen können, die durch mindestens einen Whitespace (' ', '\t', '\n', '\r') voneinander getrennt sind. Das Ergebnis soll in einer weiteren Datei *ergebnis.txt* abgelegt werden.

Überprüfen Sie dabei auch, ob das Öffnen gutgeht (und vergessen Sie nicht, die Datei nach Ende der Bearbeitung zu schließen!).

Konnte die Datei nicht geöffnet werden, dann geben Sie einen Fehlerhinweis aus². Andernfalls können Sie mit `fscanf()` weiterarbeiten.

2. Anzahl der Zeichen (blatt8_2.c)

Implementieren Sie ein Programm, das eine von Ihnen erstellte Textdatei einliest (z.B. *test.txt*) und dann die Anzahl aller in der Datei vorkommenden Zeichen ausgibt.

Zudem soll dabei gleichzeitig das Auftreten der verschiedenen Vokale (a, e, i, o, u) gezählt und am Ende aufgeschlüsselt nach dem jeweiligen Vokal ausgegeben werden, wobei für die Zählung Groß- und Kleinschreibung egal sein soll (verwenden Sie dazu am besten ein `switch!`).

Tipp: Die Datei sollte zunächst möglichst einfach gehalten sein, damit Sie ihr berechnetes Ergebnis durch Nachzählen leicht überprüfen können.

3. Palindrom (blatt8_3.c)

Schreiben Sie eine Funktion `bool palindrom (char str[])`, die testet, ob ein übergebenes Wort ein Palindrom ist und das Ergebnis entsprechend zurückgibt. Groß- und Kleinschreibung soll hierfür nicht beachtet werden. Dafür darf innerhalb der `palindrom`-Funktion gerne der String erst mit einer Hilfsfunktion in nur Kleinbuchstaben (oder nur Großbuchstaben) umgewandelt werden. Testen Sie die Funktion ausgiebig in der `main`-Funktion.

4. Programm-Argumente (blatt8_4.c)

Informieren Sie sich, wie in Ihrer IDE beim Ausführen des Programms Argumente mit übergeben werden können, wie z.B. bei Verwendung der Konsole über: `./test arg1 arg2, o.ä.`

Schreiben Sie dann ein Programm, das mit drei Argumenten aufgerufen werden soll: Einem Tag (1-31), einem Monat (Januar-Dezember - nicht als Zahl!) und einem Jahr. Prüfen Sie in Ihrem Code, ob es sich um ein gültiges Datum handelt, welches bereits vergangen ist, d.h. nicht in der Zukunft liegt. Zur Einfachheit dürfen Sie annehmen, dass der 31.Dezember 2025 ist, d.h. erst Daten ab dem Jahr 2026 gelten als Zukunft.

Falls es sich um ein gültiges Datum handelt, geben Sie es in der Form YYYY-MM-DD aus.

Hinweis 1: Geben Sie zu Beginn alle Argumente nacheinander aus, um zu verstehen, in welcher Form Sie im Programm ankommen.

Hinweis 2: Die Funktion `int atoi(const char *str)` aus der `stdlib.h`-Bibliothek kann Zahlen aus einer Zeichenkette in Integer umwandeln.

¹Hinweis: Fügen Sie ggfs. noch folgende Include-Direktive hinzu: `#include <limits.h>`. Damit stehen Ihnen verschiedene nützliche Konstanten zur Verfügung, wie etwa der kleinste Integerwert `INT_MIN` oder der größte Integer-Wert `INT_MAX`.

²Die Datei zum Einlesen muss im gleichen Ordner sein, wie der Quelltext, da dort zuerst gesucht wird. Alternativ kann man statt dem Dateinamen auch den gesamten Pfad angeben.

5. Hochladen und Vorstellen

Stellen Sie Ihre Lösung am Montag, den 15. Dezember 2025, in Ihrer Übungsgruppe vor. Laden Sie bis spätestens Dienstag, den 16. Dezember 2025, 23:59 Uhr, die Dateien `blatt8_1.c`, `blatt8_2.c`, `blatt8_3.c` und `blatt8_4.c` im eLearning hoch. Überprüfen Sie, dass die erreichten Punkte auch als Bewertung für Ihre Abgabe eingetragen werden. Melden Sie sich ansonsten zeitnah, damit dies nachgeholt werden kann.