

Übung 5

a) Schreiben Sie ein Programm, welches das Minimum von zehn positiven ganzen Zahlen berechnet und ausgibt. Dabei sollen so lange Zahlen von der Konsole eingelesen werden, bis man zehn positive Zahlen gesehen hat, von welchen dann das Minimum bestimmt wird. Negative Zahlen und Null werden einfach ignoriert.

b) Berechnen Sie das Maximum aus einer Menge von Ganzzahlen (auch negative), welche in einer selbsterstellten 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 aus.² Andernfalls können Sie mit *fscanf()* weiterarbeiten.

Hinweis: Meistens haben sich in der ersten Version eines Programms noch einige Fehler eingeschlichen, wodurch es nicht richtig funktioniert. Die Ausgabe von Variablenwerten an ‚neuralgischen‘ Stellen eines Programms (sog. Logging) erleichtert das Debuggen enorm (also das Finden und Korrigieren von Bugs, d.h. Fehlern).

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 gezählt und am Ende aufgeschlüsselt nach dem jeweiligen Vokal ausgegeben werden, wobei für die Zählung Groß- und Kleinschreibung egal sein soll (nutzen Sie dazu am besten einen *switch*).

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

d) Die Datei *code_test.c* auf Moodle soll die ganzzahlige Division berechnen, d.h. bei Eingabe von 17 und 4 soll 17 / 4 berechnet und vier ausgegeben werden. Testen und debuggen Sie dieses Programm. Notieren Sie sich, welche Testfälle Sie benötigen. (Es ist mindestens ein Fehler zu finden und zu beheben.)

Überlegen Sie dabei auch, ob Logging, also Ausgaben auf die Konsole, „scharfes Hinsehen“ oder Schritt-für-Schritt-Verfolgung im Debugger (→ Recherchieren)³ Ihnen am ehesten half, den ursächlichen Fehler zu finden.

¹ Hinweis: 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 Wert `INT_MAX`.

² Wenn man in MS Visual Studio ein C-Programm klassisch über den Debugger startet, dann ist der ausführende Pfad, also der Pfad, wo die Datei zuerst gesucht wird, das Verzeichnis in dem der Quelltext liegt. (Startet man direkt über die Konsole, wird im aktuellen Pfad gesucht.) Alternativ kann man statt dem Dateinamen auch den gesamten Pfad angeben.

³ S. <https://docs.microsoft.com/de-de/visualstudio/debugger/debugger-feature-tour?view=vs-2022>