

**Ingenieurinformatik (FK 03)**  
**Übung 3**

### VORBEREITUNG

Erstellen Sie die Struktogramme für **prim1** und **prim2** der Übung 3b mithilfe des Programms **Structorizer**.

Machen Sie sich mit dem Debugger in QT Creator vertraut  
(s. [http://pc01-lsw.ee.hm.edu/wicked/QTC\\_Debugging](http://pc01-lsw.ee.hm.edu/wicked/QTC_Debugging)).

### ÜBUNG 3A (FEHLERBEHEBUNG, FEHLERANALYSE)

#### Einführung

Bei der Softwareentwicklung gibt es unterschiedliche Fehlerarten:

- Fehler in der Problemanalyse
- Fehler im Lösungsalgorithmus

Hier wird typischerweise das Problem unvollständig oder falsch beschrieben → Überprüfung des Struktogramms, Abgleich des Codes mit der Algorithmusbeschreibung, Erhöhung der Lesbarkeit des Codes durch Umformatierung.

- Übersetzungsfehler oder –warnungen  
Typische Fehler sind:  
*Fehlendes Semikolon oder andere Syntaxfehler*  
*Anführungsstriche nicht abgeschlossen*  
*Groß-/Kleinschreibung nicht beachtet*  
*Variable wird verwendet, wurde aber nicht deklariert/definiert*  
*Ausdruck fehlerhaft*  
*Schreibfehler*  
*(runde oder geschweifte) Klammer vergessen oder zuviel.*
- Laufzeitfehler  
Programm wird fehlerfrei übersetzt, aber während der Laufzeit treten Fehler auf.  
Typische Fehler sind:  
*Ignorieren von Warnungen*  
*Verwendung von nichtinitialisierten Variablen*  
*Überschreitung des Wertebereichs der Variablen*  
*Nichtbeachten fehlerhafter Eingaben des Benutzers*  
*Kontrollstrukturen werden nicht durchlaufen oder nie verlassen*  
    → *Endlosschleifen*  
    → *fehlerhafte Berechnungen/Ausführung des Teilalgorithmus*  
*Fehlender Adressoperator (&) bei scanf ()*  
*Fehlende Überprüfung fehlerhafter Eingaben*  
    → *Division durch 0*  
    → *Speicherfehler*

#### Ziel

Es sollen diverse Beispiele analysiert, die Lesbarkeit des Codes erhöht werden und eventuelle Fehler behoben werden.

## Durchführung

- Starten Sie das Programm **QT Creator**.
- Erstellen Sie in diesem Arbeitsbereich ein **C-Projekt** mit dem Namen **div**
- Unter dem Verzeichnis **U: \workspace**
- Kopieren Sie nun die bereitgestellte Datei **div.c** in das Verzeichnis **U: \workspace\div** mithilfe des Windows-Explorer.
- Fügen Sie die Datei dem Projekt hinzu und entfernen die bereitgestellte Datei **main.c**.
- Das Programm berechnet die Ganzzahldivision zweier eingegebener Zahlen und den Rest der dabei entsteht.  
Sollte der Divisor 0 sein, dann wird das Programm mit einer Textausgabe beendet **ohne** die Berechnungen durchzuführen.  
Beheben Sie die Fehler im bereitgestellten Quellcode, so dass die Funktion gewährleistet ist und der Quellcode sich fehler- und warnungsfrei erstellen läßt.
- **Frage A1:** Welche Fehler haben Sie gefunden und um welche Fehlertypen handelt es sich?
  
- Schließen Sie ggf. bereits geöffnete Projekte und erstellen Sie ein neues **C-Projekt** mit dem Namen **qsum**.
- Kopieren Sie nun die bereitgestellte Datei **qsum.c** in das Verzeichnis **U: \workspace\qsum** mithilfe des Windows-Explorer.
- Fügen Sie die Datei dem Projekt hinzu und entfernen die bereitgestellte Datei **main.c**.
- Versuchen Sie das bereitgestellte Programm, dass die Summe der Quadrate von 1 .. 50 ermittelt, zu erstellen und auszuführen.
- Beheben Sie die Fehler, bis das Programm die korrekte Ausgabe liefert:  
**Summe = 42925**
- **Hinweis:** Zur Fehlersuche ist es manchmal sinnvoll Zwischenergebnisse mithilfe von **printf** auszugeben.
- **Frage A2:** Welche Fehler haben Sie gefunden und um welche Fehlertypen handelt es sich?
  
- Schließen Sie ggf. bereits geöffnete Projekte und erstellen Sie ein neues **C-Projekt** mit dem Namen **test1**.
- Kopieren Sie nun die bereitgestellte Datei **test1.c** in das Verzeichnis **U: \workspace\test1** mithilfe des Windows-Explorer.
- Fügen Sie die Datei dem Projekt hinzu und entfernen die bereitgestellte Datei **main.c**.
- Erhöhen Sie die Lesbarkeit des Programms indem Sie für neue Verschachtelungsebenen geeignet einrücken (entweder 4 Leerzeichen oder ein Tabulator).  
*Siehe Quellcodebeispiele in Kapitel 4 der Vorlesungsunterlagen.*
- Versuchen Sie das bereitgestellte Programm zu erstellen.
- Starten Sie das Programm.
- **Frage A3:** Was macht das Programm?
- Ändern Sie die Variablennamen von **a** und **b** in aussagekräftigere Variablennamen um.
- Erstellen Sie nun das Programm erneut und starten es aus der Eingabeaufforderung.
- **Frage A4:** Wie werden die anderen zwei Programme über die Eingabeaufforderung gestartet?

## ÜBUNG 3B (PRIMZAHLENERMITTLUNG)

### Einführung

Als Primzahlen werden positive Ganzzahlen ( $>1$ ) bezeichnet, die nur durch 1 oder sich selbst teilbar sind.

Zur Ermittlung dieser Zahlen wird folgender Algorithmus verwendet:

- Erfrage vom Benutzer den Maximalwert (Variable **max\_wert**) bis zu dem die Primzahlen ermittelt werden sollen
- Äußere Schleife: Variable **n** läuft von 2 bis **max\_wert**
  - Innere Schleife: Variable **teiler** läuft von 2 bis **n-1**
    - Jeweils prüfen, ob **n** durch **teiler** teilbar ist.  
Falls ja, kann Wert in **n** keine Primzahl sein.  
*Setzen Sie dazu eine geeignete Variable, die diesen Fall widerspiegelt.*
  - Falls innere Schleife durchgelaufen ist, ohne einen Teiler zu finden, wird die Primzahl (Wert in **n**) ausgegeben.

### Ziel

Es sollen die unterschiedlichen Schleifenkontrollstrukturen (**do-while**, **while**, **for**) untersucht werden.

Der oben dargestellte Primzahlenalgorithmus soll auf zwei Arten realisiert werden:

- 1) **prim1**: äußere Schleife mit **do-while**, innere Schleife mit **while**
- 2) **prim2**: äußere Schleife mit **while**, innere Schleife mit **for**

### Durchführung

- Schließen Sie ggf. bereits geöffnete Projekte und erstellen Sie ein neues **C-Projekt** mit dem Namen **prim1**
- Realisieren Sie die Vorgaben Ihres Struktogramms in der Funktion **main**.
- Starten Sie Ihr Programm mit dem Debugger und durchlaufen einige Iterationen der inneren Schleife im Einzelschrittmodus.  
Beobachten Sie dabei die Änderungen der Variablen.  
Setzen Sie einen geeigneten Haltepunkt (*Breakpoint*), z. B. nach Ablauf der inneren Schleife und durchlaufen Sie einige Iterationen der äußeren Schleife.  
Eine Anleitung zum Debugger finden Sie unter:  
**[http://pc01-lsw.ee.hm.edu/wicked/QTC\\_Debugging](http://pc01-lsw.ee.hm.edu/wicked/QTC_Debugging)**
- Erstellen Sie das Programm und testen es.
- Schließen Sie ggf. bereits geöffnete Projekte und erstellen Sie ein neues **C-Projekt** mit dem Namen **prim2**.
- Realisieren Sie die Vorgaben Ihres zweiten Struktogramms in der Funktion **main**.
- Erstellen Sie das Programm und testen es.
- **Frage B1**: Wie verhalten sich die Programme bei Eingabe des Maximalwerts von 1?
- **Frage B2**: Wie kann die Laufzeit der Programme verbessert werden?