

## 3. Übung Informatik PG 1

---

In dieser Übung geht es um Kontrollstrukturen sowie Funktionen.

### 1. Kontrollstrukturen

- If/While/For: Schreiben Sie ein Programm, in dem ein eingelesener Wert mit einem "geheimen" Integer-Wert verglichen wird, eine Art Ratespiel also. (Ein ähnliches Programm haben Sie schon in der Vorlesung gesehen - Sinn der Übung ist natürlich, dass Sie das Programm selbstständig entwickeln.)
  - Sie geben den geheimen Wert vor
  - Der/die Benutzer:in wird aufgefordert, einen Wert einzugeben
  - Der eingegebene Wert wird mit dem geheimen Wert verglichen. Sind die Werte gleich oder ist der eingegebene Wert zu klein bzw. zu groß, wird ein entsprechender Hinweis ausgegeben. Sind die Werte gleich, wurde also die geheime Zahl erraten, so endet das Programm.
  - Setzen Sie den Code einmal so auf, dass das Spiel solange läuft bis die richtige Zahl geraten wurde und einmal so, dass das Spiel nach 10 Versuchen beendet wird.
- Switch: Schreiben Sie eine Switch-Anweisung, die für die verschiedenen Monate die Anzahl an Tagen ausgibt.
  - Aufgrund der Schaltjahre müssen sowohl Monat als auch Jahr abgefragt werden. Berechnung eines Schaltjahres: <https://learn.microsoft.com/de-de/office/troubleshoot/excel/determine-a-leap-year>
  - Überlegen Sie sich wie Sie ausnutzen können, dass die meisten Monate 30 bzw. 31 Tage haben.
- Schreiben Sie sich ein Programm mit zwei verschachtelten for-Schleifen mit Ausstiegswarnung (break / continue) z.B.:
  - Die erste Schleife zählt bis 20 und gibt den jeweiligen Wert aus
  - Die zweite, innere Schleife zählt ebenfalls bis 20 und gibt den Wert aus. Erreicht die innere Schleifenvariable einen Wert von 5, wird eine Ausstiegswarnung aufgerufen.
  - Welches Verhalten beobachten Sie, wenn Sie **break** verwenden? Welches Verhalten bei **continue**?

### 1. Funktionen

- Verwenden Sie das Programm `example_function_1.c` um sich mit Funktionen vertraut zu machen
  - Entfernen Sie aus dem Programm die Funktionsdeklaration der `AddNumbers`-Funktion, kompilieren Sie das Programm und schauen Sie sich die Fehlermeldungen an. In welchen Zeilen des Programms findet der Compiler die Warnungen / Fehler?
  - Nutzen Sie das Debug-Tool Ihrer IDE um den Programmablauf zu beobachten.
  - Schreiben Sie das Programm so um, dass die `addNumbers`-Funktion als Vorwärtsdeklaration deklariert wird.
  - Schreiben Sie im Programm `example_function_1.c` die `addNumbers`-Funktion so um, dass keine `Return`-Funktion verwendet wird, die grundsätzliche Funktionsweise des Programms (der Ergebniswert wird auf die Konsole ausgegeben) aber erhalten bleibt.

- Schreiben Sie das ursprüngliche Programm `example_function_1.c` so um, dass die zu addierenden Werte globale Variablen sind.
- Erweitern Sie das ursprüngliche Programm `example_function_1.c` so, dass alle Grundrechenarten abgedeckt werden (+, -, /, \*). Überlegen Sie sich, ob es sinnvoll ist alle Rechenarten in einer Funktion unterzubringen oder mehrere Funktionen zu verwenden (die Antwort hängt von Ihrer Umsetzung ab, es gibt also kein falsch oder richtig). Überlegen Sie sich, wo welche Variablen sichtbar sind (Stichwort Geltungsbereich der Variablen).
- Rückgabewert von Funktionen: Auch die `printf`-Funktion hat einen Rückgabewert. Wie können Sie diesen auslesen und was sagt dieser Wert aus?