

## Erste Schritte in C / Datentypen

### Aufgabe 1: Übersetzen in der Konsole unter Linux

Gegeben ist folgendes C-Programm:

```
1 #include <stdio.h>
2
3 int mein(void) {
4
5     int i;
6     double x=0.0;
7
8     for ( i=0 ; i<10 ; i++ ) {
9         printf {"i=%d x=%g\n",i,x};
10        x = x + i
11    }
12    return 0;
13 }
```

- Erstellen Sie eine Quellcode-Datei mit obigem C-Code.
- Übersetzen Sie dieses Programm / Entfernen Sie die eingebauten Fehler. Werden vom Compiler alle Fehler wie erwartet angezeigt?
- Messen Sie, wie lange Ihr System zum Übersetzen und Ausführen benötigt, was fällt Ihnen hierbei auf?

Hinweise:

- Sie können einen Editor Ihrer Wahl verwenden, z.B. den in den Dateimanager integrierten `gedit`, den klassischen `vi` oder `vim` oder den 'komplexen' `emacs`.
- Der C-Compiler unter Linux wird mit `gcc` (Option `-o`) oder über das Kommando `make` aufgerufen (das funktioniert auch ohne `Makefile`, aber das lernen wir später).
- Laufzeiten von Programmen können unter Linux in der Konsole mit dem vorangestellten Kommando `time` durchgeführt werden.

### Aufgabe 2: Übersetzen in der Entwicklungsumgebung

Führen Sie die ersten beiden Schritte aus Aufgabe 1 in einer 'Entwicklungsumgebung' durch.

- Sie können hierzu `CLION` unter Linux oder Windows oder eine Entwicklungsumgebung Ihrer Wahl (Eclipse, Visual Studio, CodeBlocks, Xcode, etc.) verwenden.
- Was fällt Ihnen bei der Zeitmessung (Compilation, Programmlaufzeit) auf?

### Aufgabe 3: Zahlen: int, long, float & double

In C werden ganze Zahlen und Fließkommazahlen intern auf unterschiedliche Weise repräsentiert (`int/long` als Dualzahlen, `float/double` mit einer begrenzten Anzahl Ziffern und einem separaten Exponenten zur Basis 10).

Werden Integer-Zahlen oder Double-Zahlen immer wieder (z.B. in einer Schleife) um einen konstanten Wert (z.B. `1` bzw. `1.0`) erhöht, werden:

- positive `int` oder `long int` Zahlen plötzlich negativ,
- `float/double` Zahlen verändern Ihren Wert nicht mehr.

Schreiben Sie ein C-Programm, das dieses Verhalten zeigt. Also:

- die größte positive `int` Zahl bestimmt und ausgibt,
- diejenige `float` Zahl ausgibt, die sich bei Addition um `1.0` aufgrund von Rundungsfehlern nicht mehr verändert.

Sie können sich diesen Zahlen durch 'geschicktes Raten' oder durch programmierte Schleifen nähern.

Würde Ihr Programm auch bei `double` Variablen funktionieren?