

# PROGC Selbststudium 1 - Kontrollstrukturen

## Inhalt

PROGC Selbststudium 1 - Kontrollstrukturen .....	1
1 Konvertierung Celsius ↔ Fahrenheit.....	1
1.1 Aufgabe.....	1
1.2 Hinweise .....	1
2 Zählen von Zeichen und Wörtern .....	1
2.1 Aufgabe.....	1
2.2 Hinweise .....	2
3 Testen der Aufgaben.....	2
3.1 Hinweise .....	2

## 1 Konvertierung Celsius ↔ Fahrenheit

### 1.1 Aufgabe

Schreiben Sie ein Programm in C, welches für die Temperaturwerte von -100 bis +200 Grad Fahrenheit (in 20er Schritten) die entsprechenden Werte in Celsius ausgibt. Die Ausgabe soll in einer sauber formatierten Tabelle analog zum folgenden Beispiel erscheinen (Celsius auf zwei Stellen gerundet, vier Leerzeichen zwischen F'heit und Celsius):

F'heit	Celsius
-----	-----
-100	-73.33
-80	-62.22
	.
	.
	.
160	71.11
180	82.22
200	93.33

### 1.2 Hinweise

- Die Formel für die Umrechnung von Fahrenheit (f) in Celsius (c) Werte lautet:

$$c = \frac{5 \cdot (f - 32)}{9}$$

- Für die Aufgabe brauchen Sie eine Schleife und die **printf**-Funktion. Siehe die verschiedenen Praktika Aufgaben wo **printf** verwendet und teilweise erklärt wird.

## 2 Zählen von Zeichen und Wörtern

### 2.1 Aufgabe

Schreiben Sie ein Programm in C, welches die Zeichen und Wörter einer mit der Tastatur eingegebenen Zeile zählt. Wortzwischenräume sind entweder Leerzeichen ( ' ') oder Tabulator-

zeichen ( `'\t'` ). Eine Zeile wird bei der Eingabe mit einem Newline-Character ( `'\n'` ) abgeschlossen, worauf das Programm die Anzahl Zeichen (ohne dem abschliessenden Newline-Character) und Wörter ausgeben und terminieren soll.

## 2.2 Hinweise

Lesen Sie die einzelnen Zeichen mit der Funktion `int getchar(void)` aus der `stdio.h` Library ein. Die Funktion `getchar` kehrt nicht gleich bei Eingabe des ersten Zeichens zurück, sondern puffert die Daten, bis die Eingabe einer kompletten Zeile mit Return abgeschlossen wird. Dann wird das erste Zeichen aus dem Puffer zurückgegeben und mit weiteren Aufrufen von `getchar` können die nachfolgenden Zeichen aus dem Puffer gelesen werden. Gibt `getchar '\n'` zurück, ist die Zeile komplett zurückgegeben und der Puffer ist wieder leer.

## 3 Testen der Aufgaben

Verwenden Sie wie in den Praktika das `CUnit` Test Framework.

In den verschiedenen Praktika sind zwei Arten von Tests Implementationen zu sehen:

- 1) Tests welche das **Programm** starten (**TARGET**), Kommandozeilen-Argumente angeben und den Standard-Out und -Error je in ein separates File umleiten. Der Exit Code wird dort geprüft plus der erwartete Output mit dem effektiven Output verglichen. Das ist nützlich wenn man `printf` Output und Error Handling prüfen will.  
Beispiel Praktika: `lab01-hello-world`, `lab03-modular`, etc.
- 2) Eigentliche Unit Tests im engeren Sinn: **spezifische Funktionen** werden im Test direkt aufgerufen und z.B. der Return Wert geprüft.  
Beispiel Praktika: `lab06-pointer-arithmetic`.

### 3.1 Hinweise

- Sie können z.B. das Praktikumsprojekt `lab01-hello-world` als Muster (Makefile anpassen, etc.).
- Wenn sie Funktionen aus der Math Library verwenden, müssen Sie die Linker Aufrufe im Makefile entsprechend ergänzen. Siehe dazu die entsprechende *man* Page der verwendeten Funktion. Dort steht wie zu linken ist.