

# Programmieren 1 - WS 2022/23

Prof. Dr. Michael Rohs, Tim Dünke, M.Sc., Jan Feuchter, M.Sc.

## Präsenzübung 5

Diese Aufgaben sind zur Lösung während der einstündigen Präsenzübung gedacht. Sie können die Aufgaben auf einem mitgebrachten Laptop oder auf Papier lösen.

### Aufgabe 1: Werte, Variablen, Zuweisungen, Deklarationen

- Grenzen Sie die Begriffe *Wert* und *Variable* voneinander ab.
- Beschreiben Sie was bei einer *Zuweisung* passiert. Nutzen Sie Ihr Wissen über Werte und Variablen. Geben Sie ein Beispiel an.
- Wozu dient die Deklaration einer Variablen in C? Geben Sie ein Beispiel für eine Variablendeklaration und ein Beispiel für eine Funktionsdeklaration in C. Welche Fehler können durch Deklarationen vermieden werden, z.B. im Vergleich zu Sprachen ohne Variablendeklaration, wie PostFix.

### Aufgabe 2: Score für Namen

Schreiben Sie eine Funktion in C, die für einen gegebenen Namen (z.B. Vorname oder Familienname) einen Score berechnet. Die Template-Datei ist `name_score.c`. Der Einfachheit halber sollen nur die Buchstaben A bis Z im Namen vorkommen. Umlaute und Eszett müssen also ggf. aufgelöst und Akzente weggelassen werden. (Eine spätere Aufgabe wird diese Einschränkung aufheben.) Großbuchstaben können mit der Funktion `int tolower(int c)` in Kleinbuchstaben verwandelt werden. Die Anzahl der Zeichen in einer Zeichenkette (String) kann mit `int s_length(String s)` bestimmt werden. Auf ein bestimmtes Zeichen kann mit `char s_get(String s, int i)` zugegriffen werden. Ein String-Literal wird als `"abc"` geschrieben. Ein einzelnes Zeichen wird als `'a'` geschrieben und steht für die Position des Zeichens in der ASCII-Tabelle. Z.B. gilt `'a' == 97`. Daher können Zeichen auch verglichen werden, z.B. `'a' < 'b'`. Mit `putc(c)` wird das Zeichen als Zeichen ausgegeben, mit `puti(c)` wird das Zeichen als Zahl (z.B. 97) ausgegeben.

Der Score soll nach folgenden Regeln gebildet werden:

- Jeder Vokal ergibt einen Punkt.
  - Jedes j, p und q ergibt fünf Punkte.
  - Jeder Buchstabe ab einschließlich v (bis zum Ende des Alphabets) ergibt drei Punkte.
  - Andere Buchstaben werden nicht gewertet.
  - Wenn im Namen ein anderes Zeichen als a-z auftaucht, ist der Gesamtscore -1.
  - Klein- und Großbuchstaben werden gleich behandelt.
- Schreiben Sie einen Kopfkomentar (purpose statement) für die Funktion `score`.
  - Implementieren Sie die Funktion `score`.
  - Prüfen Sie die Testfälle.

### Aufgabe 3: Schleifen

In dieser Aufgabe sollen verschiedene Varianten von Schleifen implementiert werden. Die Template-Datei ist `loops.c`.

- a) Verwandeln Sie die `while`-Schleife in eine äquivalente `for`-Schleife.
- b) Verwandeln Sie die `while`-Schleife in eine äquivalente `for`-Schleife.
- c) Verwandeln Sie die `while`-Schleife in eine äquivalente `for`-Schleife.
- d) Implementieren Sie die Funktion `void print_outer_row(int width)`, die eine Zeile der folgenden Art ausgibt:
  - für `width = 4`: `+---+`
  - für `width = 0`: `+`
  - für `width = 1`: `++`
  - für `width = 2`: `+-+`
- e) Implementieren Sie die Funktion `void print_inner_row(int width)`, die eine Zeile der folgenden Art ausgibt:
  - für `width = 4`: `|---|`
  - für `width = 0`: `|`
  - für `width = 1`: `||`
  - für `width = 2`: `|-|`
- f) Implementieren Sie die Funktion `void rect(int width, int height)`, die ein Rechteck der folgenden Art ausgibt. Die Funktion soll die Hilfsfunktionen `print_inner_row` und `print_outer_row` verwenden.
 

```
rect(4, 3);
+----+
|    |
|    |
|    |
+----+
```
- g) Die Funktionen `print_inner_row` und `print_outer_row` sind sehr ähnlich. Schreiben Sie eine Funktion `print_row(char outer, char inner, int width)`, die beide Funktionen generalisiert. Implementieren Sie diese Funktion dann als Hilfsfunktion zur Neuimplementierung von `print_inner_row` und `print_outer_row`.