

# Hardwarenahe Programmierung / Angewandte Informatik

## Übungsaufgaben – 14. November 2016

### Aufgabe 1: Fakultät

Die Fakultät  $n!$  einer ganzen Zahl  $n \geq 0$  ist definiert als:

$$\begin{aligned} &1 \quad \text{für } n = 0, \\ &n \cdot (n - 1)! \quad \text{für } n > 0. \end{aligned}$$

Mit anderen Worten:  $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$ .

Die folgende Funktion `fak()` berechnet die Fakultät *rekursiv* (Datei: [aufgabe-1.c](#)):

```
int fak (int n)
{
    if (n <= 0)
        return 1;
    else
        return n * fak (n - 1);
}
```

- (a) Schreiben Sie eine Funktion, die die Fakultät *iterativ* berechnet, d. h. mit Hilfe einer Schleife anstelle von Rekursion.
- (b) Wie viele Multiplikationen (Landau-Symbol) erfordern beide Versionen der Fakultätsfunktion?
- (c) Wieviel Speicherplatz (Landau-Symbol) erfordern beide Versionen der Fakultätsfunktion?

### Aufgabe 2: Strings

Wir betrachten nochmals die Funktion aus der vorherigen Übung (Datei: [aufgabe-2.c](#)):

```
int fun_1 (char *s1, char *s2)
{
    int result = 1;
    for (int i = 0; s1[i] && s2[i]; i++)
        if (s1[i] != s2[i])
            result = 0;
    return result;
}
```

- (e) Von welcher Ordnung (Landau-Symbol) ist die Funktion `fun_1()` hinsichtlich der Anzahl ihrer Zugriffe auf die Zeichen in den Strings – und warum?
- (f) Von welcher Ordnung (Landau-Symbol) ist die von Ihnen in Aufgabenteil (d) geschriebene effizientere Version der Funktion?

### Aufgabe 3: Text-Grafik-Bibliothek

Ergänzen Sie die Text-Grafik-Bibliothek aus der letzten Übung um eine weitere Funktion:

- **`void fill (int x, int y, char c, char o)`**  
Fläche in der „Farbe“ `c`, die den Punkt `(x, y)` enthält, mit der „Farbe“ `c` ausmalen

Hinweise:

- Führen Sie eine Web-Recherche nach dem Begriff „Floodfill“ durch.
- Schreiben Sie ein Test-Programm, das interessante umrandete Flächen – auch mit „Loch“ – „ausmalt“.