

# Angewandte Informatik

## Übungsaufgaben – 19. November 2015

### Aufgabe 1: Fakultät

Die Fakultät  $n!$  einer natürlichen Zahl  $n$  ist definiert als:

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

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

Die folgende Funktion `fak()` berechnet die Fakultät *rekursiv*:

```
#include <stdio.h>

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

int main (void)
{
    for (int n = 1; n <= 5; n++)
        printf ("%d\n", fak (n));
    return 0;
}
```

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

## Aufgabe 2: Kalender-Berechnung

Am 3. 1. 2009 meldete *heise online*:

Kunden des ersten mobilen Media-Players von Microsoft erlebten zum Jahresende eine böse Überraschung: Am 31. Dezember 2008 fielen weltweit alle Zune-Geräte der ersten Generation aus. Ursache war ein interner Fehler bei der Handhabung von Schaltjahren.

<http://heise.de/-193332>,

Der Artikel verweist auf ein Quelltextfragment, das für einen gegebenen Wert `days` das Jahr und den Tag innerhalb des Jahres für den `days`-ten Tag nach dem 1. 1. 1980 berechnen soll:

```
year = ORIGINYEAR; /* = 1980 */

while (days > 365)
{
    if (IsLeapYear (year))
    {
        if (days > 366)
        {
            days -= 366;
            year += 1;
        }
    }
    else
    {
        days -= 365;
        year += 1;
    }
}
```

Dieses Quelltextfragment weist mehrere Code-Verdopplungen auf:

- Die Anweisung `year += 1` taucht an zwei Stellen auf.
- Es gibt zwei unabhängige Abfragen `days > 365` und `days > 366`: eine in einer **while**- und die andere in einer **if**-Bedingung.
- Die Länge eines Jahres wird nicht durch eine Funktion berechnet oder in einer Variablen gespeichert; stattdessen werden an mehreren Stellen die expliziten numerischen Konstanten 365 und 366 verwendet.

Diese Probleme führten am 31. Dezember 2008 zu einer Endlosschleife, die sich – z. B. durch eine Funktion `DaysInYear()` – leicht hätte vermeiden lassen.

Gut hingegen ist die Verwendung einer Präprozessor-Konstanten `ORIGINYEAR` anstelle der Zahl 1980 sowie die Kapselung der Berechnung der Schaltjahr-Bedingung in einer Funktion `IsLeapYear()`.

- (a) Erklären Sie das Zustandekommen der Endlosschleife.
- (b) Schreiben Sie das Quelltextfragment so um, daß es die beschriebenen Probleme nicht mehr enthält.