Dr. habil. Bernd Schürmann
Dr. Annette Bieniusa
pinc-support@cs.uni-kl.de

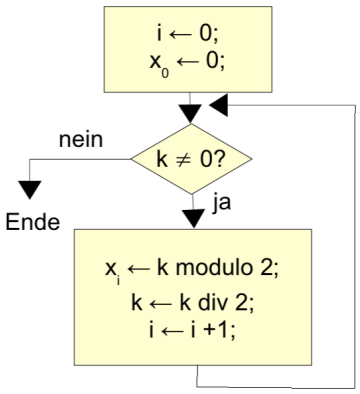
TU Kaiserslautern
Fachbereich Informatik

Lösungshinweise zum Übungsblatt 5: Programmieren in C (WS 2018/19)

1. Dieses Übungsblatt ist ein **Pflichtübungsblatt**, d.h. Sie müssen bei der Bearbeitung dieses Blattes **mind. 50% der Gesamtpunkte** erreichen.
2. Die Abgabe erfolgt über das Exclaim-System gemeinsam mit Ihrem Teampartner.
3. Zur Beantwortung von Fragen und Hilfestellung bei der Bearbeitung kommen Sie bitte in die wöchentliche Fragestunde, Mittwoch 15:30, in Terminalraum 32-410.
4. Bitte laden Sie einzelne Dateien hoch, wie in der Aufgabenstellung angegeben, keine Archive, keine kompilierten Dateien.
5. Programme, die nicht kompilieren, werden nicht korrigiert und mit **0 Punkten** bewertet.
6. Bitte beachten Sie unsere Regelung bei Plagiaten:
   - Wenn Sie sekundäre Quellen, wie Bücher oder das Internet verwenden, müssen Sie immer die Quelle angeben. Das einfache Kopieren aus anderen Quellen ist für die Übungen nicht gestattet. Wenn wir in einer Übungsabgabe kopierten Code ohne Quellenangabe finden, wird die gesamte Abgabe mit 0 Punkten bewertet.
   - Sie können Übungsaufgaben gerne mit den Mitgliedern anderer Teams diskutieren. Sie sollten jedoch Ihren Code vor der Abgabefrist nicht an andere Teams weitergeben bzw. zeigen.
   - Wenn Code von anderen Teams kopiert wurde, werden die Abgaben **von beiden Teams** mit 0 Punkten bewertet.
   - Wir behalten uns vor Punkte auch nachträglich abzuziehen, wenn ein Verstoß erst später bemerkt wird.

Aufgabe 1: Zahlenkonvertierung I (5 Punkte, Abgabe in Exclaim als `binaer.c`)

In der Vorlesung wurde folgender Algorithmus zur Umrechnung von Dezimal- in Dualzahlen vorgestellt.



Ergänzen Sie den Algorithmus (Flussdiagramm) um die Ein- und Ausgabe und implementieren Sie den Algorithmus dann als C-Programm.

Die Ausgabe soll in Tabellenform analog zu der Tabelle auf den Vorlesungsfolien erfolgen. Das heißt, bei der Eingabe der Zahl 14 soll die Bildschirmausgabe wie folgt aussehen:

```
   k   k modulo 2   k div 2    i
================================
  14            0         7    0
   7            1         3    1
   3            1         1    2
   1            1         0    3
   0  -> Ende
```

Für die Ausrichtung der Tabelleneinträge verwenden Sie die formatierte Ausgabe von Integern mittels `printf()`. Die Anweisung `printf("%4d", x)` sorgt beispielsweise dafür, dass der Wert von `x` immer mit 4 Zeichen Breite dargestellt wird. Potentiell fehlende Ziffern werden dabei vorne mit Leerzeichen aufgefüllt.

```c
// Umwandlung in Binaerdarstellung mit Zwischenschritten

#include <stdio.h>

int main(void)
{
    int k;
    scanf("%d",&k);

    int x = 0;
    int i = 0;

    // Ausgabe Header
    printf("   k   k modulo 2   k div 2    i\n");
    printf("================================\n");

    while(k != 0) {
        printf("%4d",k);
        x = k % 2;
        printf("%13d",x);
        k = k / 2;
        printf("%10d",k);
        printf("%5d\n",i);
        i++;
    }

    // Ausgabe Ende
    printf("   0  -> Ende\n");

    return 0;
}
```

Aufgabe 2: Zahlenkonvertierung II (5 Punkte, Abgabe in Exclaim als `oktal.c`)

Schreiben Sie ein Programm, das eine positive ganze Zahl in Oktal-Darstellung als Eingabe nimmt und die entsprechende Dezimalzahl ausgibt. Verwenden Sie beim Einlesen **nicht** `scanf("%o",...)`, sondern implementieren Sie einen geeigneten Algorithmus für die Umwandlung selbst.

Beispiel: Bei der Eingabe von `71` soll die Ausgabe `57` sein.

```c
// Umwandlung von Oktal- in Dezimalzahl

#include <stdio.h>

int main(void)
{
    int x;
    scanf("%d",&x);

    int result = 0;
    int factor = 1;
    while(x > 0)
    {
```

```
        int d = x % 10;
        result += factor * d;
        x /= 10;
        factor *= 8;
    }
    printf("%d\n", result);
    return 0;
}
```

Aufgabe 3: Aufsteigende Folge von Zahlen (5 Punkte, Abgabe in Exclaim als `aufsteigend.c`)

Schreiben Sie ein Programm, das als erstes eine positive ganze Zahl `n` einliest, dann `n` Integer einliest und als Ergebnis **`aufsteigend`** ausgibt, falls die n Zahlen aufsteigend geordnet sind, sonst **`nicht aufsteigend`**.

Beispiel: Bei der Eingabe von

```
5 -10 2 6 6 29
```

soll die Ausgabe `aufsteigend` sein.
Bei der Eingabe von

```
4 1 0 6 8
```

soll die Ausgabe `nicht aufsteigend` sein.

Mit den Sprachelementen, die bisher in der Vorlesung behandelt wurde, ergibt sich z.B. folgende Lösung:

```
// Test, ob Zahlen aufsteigend sortiert sind

#include <stdio.h>

int main(void)
{
    int n, i, next, curr;
    scanf("%d", &n);

    scanf("%d",&curr);
    i = n - 1;
    while (i > 0) {
        scanf("%d", &next);
        if (curr <= next) {
            curr = next;
            i--;
        }
        else
        {
            printf("nicht aufsteigend\n");
            return 0;
        }
    }

    printf("aufsteigend\n");
    return 0;
}
```

Durch Verwendung von boolschen Variablen (hier über die Bibliothek `stdbool.h`) kann man den Test, ob die Zahlenfolge aufsteigend sortiert ist, von der Ausgabe des Ergenisses trennen. Dies führt in der Regel zu Code, der einfacher zu verstehen ist und die Wiederverwendbarkeit von Elementen erlaubt.

```
// Test, ob Zahlen aufsteigend sortiert sind

#include <stdio.h>
#include <stdbool.h>
```

```c
int main(void)
{
    int n, i, next, curr;
    bool aufsteigend;
    scanf("%d", &n);

    scanf("%d",&curr);

    i = n - 1;
    aufsteigend = true;
    while (i > 0) {
        scanf("%d", &next);
        if (curr <= next) {
            curr = next;
            i--;
        }
        else
        {
            aufsteigend = false;
            break;
        }
    }

    if(aufsteigend)
    {
        printf("aufsteigend\n");
    }
    else
    {
        printf("nicht aufsteigend\n");
    }
    return 0;
}
```

Aufgabe 4: Wochentage (5 Punkte, Abgabe in Exclaim als `wochentag.c`)

Erstellen Sie ein C-Programm, welches den Wochentag für ein bestimmtes Datum ausgibt. Dabei soll das Datum über drei Eingaben angegeben werden:

- Tag des Monats (1-31)
- Monat (1-12)
- Jahr (positive Zahl)

Verwenden Sie zur Berechnung folgende Formeln:

$$\begin{aligned} a &= j - (14 - m)/12 \\ b &= a + a/4 - a/100 + a/400 \\ c &= m + 12 * ((14 - m)/12) - 2 \\ w &= (t + b + (31 * c)/12) \bmod 7 \end{aligned}$$

Hierbei ist:

- w ist der Wochentag (0 = Sonntag, 1 = Montag, ..., 6 = Samstag)
- t ist der eingegebene Tag.
- m ist der eingegebene Monat.
- j ist das eingebene Jahr.

Beispiel: Bei Eingabe von `26 11 2018` soll folgende Ausgabe erfolgen:

```
Das Datum 26.11.2018 ist ein Montag.
```

```c
#include <stdio.h>

int main(void)
{
    int j, t, m, a, b, c, w;
    char* wochentag;

    // Eingaben einlesen
    scanf("%d", &t);
    scanf("%d", &m);
    scanf("%d", &j);

    a = j - (14 - m)/12;
    b = a + a/4 - a/100 + a/400;
    c = m + 12 * ((14 - m)/12) - 2;
    w = (t + b + (31 * c)/12) % 7;

    if (w == 0) wochentag = "Sonntag";
    if (w == 1) wochentag = "Montag";
    if (w == 2) wochentag = "Dienstag";
    if (w == 3) wochentag = "Mittwoch";
    if (w == 4) wochentag = "Donnerstag";
    if (w == 5) wochentag = "Freitag";
    if (w == 6) wochentag = "Samstag";

    printf("Das Datum %d.%d.%d ist ein %s.\n", t, m, j, wochentag);
    return 0;
}
```