



## Übungen zur Vorlesung Modellierung und Programmierung WS 2015–2016

### Blatt 5

#### Aufgabe 1 (6 Punkte)

Schreiben Sie ein Programm, das eine Textdatei einliest, Kleinbuchstaben nach Möglichkeit in Großbuchstaben umwandelt und den umgewandelten Text in eine zweite Datei schreibt. Daneben soll das Programm die folgenden Informationen auf den Bildschirm schreiben:

- Anzahl der gelesenen Zeichen,
- Anzahl der Zeilen,
- Anzahl der Buchstaben,
- Anzahl der Ziffern,
- Anzahl der Buchstaben, die nicht in Großbuchstaben umgewandelt wurden.

Die Namen der Ein- und Ausgabedatei sollen als Kommandozeilenargumente beim Programmstart übergeben werden. Selbstverständlich soll das Programm überprüfen, ob die Datenströme zu diesen Dateien geöffnet werden konnten und ob Fehler bei der Ein- bzw. Ausgabe aufgetreten sind.

#### Aufgabe 2 (2+2+2 Punkte)

Zur Erzeugung von Zufallszahlen existieren u.a. die folgende in `<stdlib.h>` deklarierte Funktion:  
`int random(void);`

Die Funktion liefert einen (pseudo-)zufälligen `int`-Wert zwischen 0 und `RAND_MAX`, einer in `<stdlib.h>` definierten Konstante, die üblicherweise an den Wertebereich des Datentyps `int` geknüpft ist.

Setzen Sie die Funktion `random` ein, um den folgenden Algorithmus zu implementieren:

1. Betrachte drei fixierte Punkte  $(x_1, y_1), (x_2, y_2), (x_3, y_3) \in \mathbb{R}^2$ , die nicht auf eine Gerade liegen, und setze  $i := 0, (X_0, Y_0) := (x_1, y_1)$ .
2. Bekomme eine Zufallszahl  $m$  aus der Menge  $\{1, 2, 3\}$  und setze

$$(X_{i+1}, Y_{i+1}) := [(x_m, y_m) + (X_i, Y_i)]/2, \quad i := i + 1.$$

3. Wiederhole Schritt 2, solange  $i < M$ . Das Ergebnis ist die Punktmenge  $\{(X_i, Y_i)\}_{i=1}^M$ .

- (a) Schreiben Sie ein Programm, das die Punktmenge  $\{(X_i, Y_i)\}_{i=1}^M$  für  $M = 10^6$  erzeugt und in der Textdatei `punkte.txt` speichert.

- (b) Visualisieren Sie die Punktmenge mit dem Programm `gnuplot` (`gnuplot`-Befehl: `plot "punkte.txt" with points pointtype 0`). Kurze Einführung in `gnuplot` finden Sie auf der Webseite.
- (c) Schreiben Sie ein Programm, das dieselbe Punktmenge erzeugt und in der Binärdatei `punkte.bin` speichert. Vergleichen Sie die Größen der Dateien `punkte.txt` und `punkte.bin`.

### Aufgabe 3 (2+2+2+2 Punkte)

Implementieren Sie mit Hilfe der Polynom-Struktur

```
struct polynom {
    unsigned grad;
    double *koeff;
};
```

folgende Funktionen:

- Dynamisches Erzeugen eines Polynoms, Löschen eines Polynoms.
- Einlesen der Koeffizienten von der Tastatur, formatierte Ausgabe des Polynoms als

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_0.$$

- Summe zweier Polynome, Produkt zweier Polynome. Das jeweilige Ergebnispolynom soll dynamisch erzeugt werden.
- Polynomauswertung an einer Stelle  $x$ . Implementieren Sie sowohl die naive Variante mit quadratischem Aufwand als auch die schnelle Version mit linearem Aufwand.

Testen Sie die Funktionen in einem Hauptprogramm.

**Abgabe der Lösungsvorschläge und Vorführung der praktischen Aufgabe vor dem 27.01.2016.**