6. Aufgabenblatt zu Funktionale Programmierung vom 17.11.2004, fällig: 24.11.2004

Themen: *Funktionen auf ganzen Zahlen, Zeichenreihen und Listen*

Für dieses Aufgabenblatt sollen Sie wie schon für Aufgabenblatt 2 ein **"literate"** Haskell-Script schreiben. Die im einzelnen zu entwickelnden Haskell-Rechenvorschriften zur Lösung der unten angegebenen Aufgabenstellungen sollen Sie dann für die Abgabe in einer Datei namens `Aufgabe6.lhs` ablegen. Versehen Sie wie auf den bisherigen Aufgabenblättern alle Funktionen, die Sie zur Lösung brauchen, mit ihren Typdeklarationen und kommentieren Sie Ihre Programme aussagekräftig. Benutzen Sie, wo sinnvoll, Hilfsfunktionen und Konstanten. Im einzelnen sollen Sie folgende Problemstellungen bearbeiten:

1. In der ersten Aufgabe beschäftigen wir uns mit b-adischen Zahlsystemen und Zahlen und deren Konvertierung. Den Schlüssel dazu liefert der folgende Satz: Für $b > 1$, $b \in IN$, läßt sich jede Zahl $x \in IN$ auf genau eine Weise in der Form

$$x = \sum_{i=0}^{n} d_i * b^i$$

darstellen, wobei $n \in IN \cup \{0\}$, $0 \leq d_i < b$, $d_n \neq 0$. Für $b = 10$ erhalten wir den Spezialfall der Dezimalzahlen, für $b = 2$ den der Binärzahlen und für $b = 16$ den der Hexadezimalzahlen. In der Folge betrachten wir den Ziffernvorrat $\{0, 1, 2, \ldots, 9, A, B, C, \ldots, Z\}$. Mit diesem Ziffernvorrat können wir b-adische Zahlen für Basen b mit $1 < b \leq 36$ darstellen.

Schreiben Sie eine Haskell-Rechenvorschrift `bConverse` zur Konversion einer b′-adischen Zahl in eine b″-adische Zahl mit $1 <$ b′,b″ ≤ 36. Dabei repräsentieren wir b-adische Zahlen als Zeichenreihen über dem Ziffernvorrat des jeweiligen b-adischen Systems. Im Hexadezimalsystem ist das etwa der Ziffernvorrat $\{0, 1, 2, \ldots, 9, A, B, C, D, E, F\}$. Die Funktion `bConverse` hat also die Signatur `bConverse :: String -> Int -> Int -> String`. Dabei bezeichnet das erste Argument die zu konvertierende Zahl, das zweite Argument die Basis des b′-adischen Systems, in dem das erste Argument vorliegt und das dritte Argument die Basis des b″-adischen Systems, in das das erste Argument konvertiert werden soll. Das Ein-/Ausgabeverhalten ist dann wie folgt festgelegt: Haben das zweite und dritte Argument einen Wert zwischen 2 und 36 und ist das erste Argument gültige

Repräsentation einer Zahl des durch das zweite Argument spezifizierten b′-adischen Systems, so wird das Argument in die Darstellung des durch das dritte Argument bestimmten b″-adischen Systems konvertiert; ansonsten ist das Ergebnis die Zeichenreihe `Ungueltige Eingabe`.

2. In dieser Aufgabe betrachten wir wieder das Szenario aus Aufgabe 3 von Blatt 5 zur Manipulation von Fließtext. Dabei ergänzen wir die Funktion `fragment` um zwei Funktionen `cut` und `paste` mit den Signaturen `cut :: [String] -> (Int,Int) -> Int -> Int -> [String]` und `paste :: [String] -> [String] -> (Int,Int) -> [String]`. Die Parameter der Funktion `cut` haben dabei dieselbe Bedeutung wie die entsprechenden Parameter der Funktion `fragment`. Allerdings schneidet die Funktion `cut` den Text aus dem Argumentfließtext heraus. Das Resultat der Funktion `cut` enthält dann den nach dem Ausschneiden verbleibenden Fließtext. Die Funktion `paste` fügt in den durch das erste Argument gegebenen Fließtext den durch das zweite Argument gegebenen Fließtext ein. Dabei kommt das 1. Zeichen des einzufügenden Fließtexts an der durch das dritte Argument bezeichneten Position zu stehen. Die Spalten des einzufügenden Fließtexts bleiben beim Einfügen erhalten, d.h. Zeichen, die im einzufügenden Fließtext untereinander stehen, stehen auch nach dem Einfügen untereinander. Dazu ist es u.U. nötig, Zeilen des Fließtexts, in den eingefügt wird, mit Leerzeichen aufzufüllen und neue Zeilen am Anfang oder am Ende zu ergänzen, etwa dann, wenn die Einfügeposition "außerhalb" des Fließtextes liegt.

3. *Alea iacta est*, *der Würfel ist gefallen*, sagte Caesar als er den Rubikon überschritt. Und sicher hätte er sich wiederholt, wäre er des nachstehenden Textes ansichtig geworden. Alle anderen, denen sich nicht wie Caesar der Sinn sogleich erschließt, bleiben Blut, Schweiß und Tränen nicht erspart, um die Punkte einzusammeln. Caesar-codiert, Würfelcodiert, Gürtel ist gut, Hosenträger sind gut, Gürtel und Hosenträger sind besser, Caesar- gefolgt von Würfelcodierung oder umgekehrt, Kennzahlen und Verschiebungen zwischen 1 und 6???

   Vielleicht hilft funktionale Programmierung, den Suchraum zu durchmustern und die Angabe zu entziffern...

   ```
   tjjfuubnbejsofofjecosjujifftducujtypjdsidcfgjfeivjdfbjsomj
   jhtgeeedswihfeffumdijbiffvpdffgfijvdfbefmtdubvjpvpsfofeofh
   ijfcsojoolmvfafigijcfbewijcsfeosjhnsjussegsfftoltojiofofaf
   icucvpstfdejhtgfjtffgffguo
   ```