## 3. Aufgabenblatt zu Funktionale Programmierung vom 12.11.2003, fällig: 19.11.2003

Themen: *Funktionen auf Tupeln und Listen*

Für dieses Aufgabenblatt sollen Sie Haskell-Rechenvorschriften zur Lösung der im folgenden angegebenen Aufgabenstellungen entwickeln und für die Abgabe in einer Datei namens `Aufgabe3.hs` in Ihrem Homeverzeichnis ablegen. Kommentieren Sie Ihre Programme wieder aussagekräftig und benutzen Sie, wo sinnvoll, Hilfsfunktionen und Konstanten.

- Schreiben Sie eine Haskell-Rechenvorschrift `isPrime`, die überpüft, ob eine als Argument übergebene natürliche Zahl eine Primzahl ist. Falls ja, soll die Funktion `isPrime` den Wahrheitswert `True`, ansonsten `False` liefern. Die kleinste Primzahl ist 2.

- Schreiben Sie eine Haskell-Rechenvorschrift `listPrime`, die die Menge der Primzahlen zwischen den als Argumente übergebenen ganzen Zahlen (einschließlich der Argumente) bestimmt und in Form einer aufsteigenden Liste ausgibt. Getestet wird nur mit echt positiven Argumenten, wobei das erste nicht größer als das zweite ist.

- Schreiben Sie eine Haskell-Rechenvorschrift `primFaktorZerlegung`, die eine als Argument übergebene natürliche Zahl $n$, $n \geq 2$, in ihre Primfaktoren zerlegt. Die Ausgabe soll dabei eine Liste von Paaren sein, deren erste Komponente den Primfaktor und deren zweite Komponente die Anzahl dieses Primfaktors in der Primfaktorzerlegung angibt, sofern diese Anzahl von 0 verschieden ist. Dabei soll die Liste der Paare aufsteigend nach Primzahlen geordnet sein. Angewendet auf das Argument 1176 soll die Funktion `primFaktorZerlegung` also die Ausgabe $[(2,3),(3,1),(7,2)]$ liefern.

- Schreiben Sie eine Haskell-Rechenvorschrift `kgv`, die das kleinste gemeinsame Vielfache zweier als Argumente übergebenen echt positiven ganzen Zahlen bestimmt.