2. Übungsaufgabe zu
Fortgeschrittene Funktionale Programmierung
Thema: Ströme
ausgegeben: 21.05.2006, fällig: 31.05.2006

Für dieses Aufgabenblatt sollen Sie Haskell-Rechenvorschriften zur Lösung der im folgenden angegebenen Aufgabenstellungen entwickeln und für die Abgabe in einer Datei namens `AufgabeFFP2.hs` in Ihrem Homeverzeichnis ablegen, wie gewohnt auf oberstem Niveau. Kommentieren Sie Ihre Programme aussagekräftig und benutzen Sie, wo sinnvoll, Hilfsfunktionen und Konstanten.

- Schreiben Sie eine 0-stellige Haskell-Rechenvorschrift `pof2s :: [Integer]`, die den Strom der 2er-Potenzen liefert, d.h. den Strom `[1,2,4,8,16,32,...`.

  Versuchen Sie, `pof2s` rekursiv ausschließlich mit Hilfe von `map` (sowie dem Listenkonstruktor "`:`" und arithmetischen Funktionen zu definieren).

- Im folgenden ist ein Anfangsstück des Pascalschen Dreiecks dargestellt:

```
           1
         1   1
       1   2   1
     1   3   3   1
   1   4   6   4   1
 1   5   10  10  5   1
...
```

  Wir bezeichnen die Zeilen dieses Dreiecks hier als Pascaltupel.

  Schreiben Sie eine 0-stellige Haskell-Rechenvorschrift `pts :: [[Integer]]`, die den Strom der Pascaltupel liefert, d.h. den Strom `[[1],[1,1],[1,2,1],[1,3,3,1],...`.

  Versuchen Sie, `pts` rekursiv ausschließlich mit Hilfe von `zipWith` (sowie den Standardkonstruktoren und -operatoren auf Listen und arithmetischen Funktionen zu definieren).

- Sei $n \geq 1$ eine natürliche Zahl und sei $d_1 d_2 \ldots d_k$ die Folge der Dezimalziffern von n. Sei weiters $p_1, p_2, p_3, \ldots, p_k, \ldots$ die Folge der Primzahlen. Dann heißt die Zahl

$$2^{d_1} 3^{d_2} 5^{d_3} \ldots p_k^{d_k}$$

  die *Gödelzahl* von n. Z.B. ist $144 = 2^4 * 3^2$ die Gödelzahl von 42 und $400 = 2^4 * 3^0 * 5^2$ die Gödelzahl von 402.

  1. Schreiben Sie eine Haskell-Rechenvorschrift `g :: Integer -> Integer`, die positive Argumente auf ihre Gödelzahl abbildet, nicht-positive Argumente auf 0.

2. Schreiben Sie eine 0-stellige Haskell-Rechenvorschrift `gs :: [Integer]`, die den Strom der Gödelzahlen beginnend für 1 liefert, d.h. das erste Listenelement ist die Gödelzahl von 1, das zweite Listenelement die Gödelzahl von 2, usw.

- Zu Ehren von Lambert Meerten heißt eine positive natürliche Zahl n, die mit ihrer Gödelzahl übereinstimmt, *Meertenszahl*. Eine Zahl, die sich höchstens um d mit $d \in \mathbb{N}_0$ von ihrer Meertenszahl unterscheidet, nennen wir eine *d-Meertenszahl*.

  Schreiben Sie eine Haskell-Rechenvorschrift `dM :: Integer -> [Integer]`, die für nicht-negative Argumente d den Strom der d-Meertenszahlen liefert, für negative Argumente den 0-Strom liefert, d.h. den nur aus Nullen bestehenden Strom. Für $d = 0$ liefert `dM` also den Strom der Meertenszahlen.

  Hinweis: Meertenszahlen sind "selten". Für $d = 0$ kann Ihre Lösung "lange" brauchen.