

Übungen zur „Praktischen Informatik III“, WS 2006/07

Nr. 2, Abgabe: 31. Oktober 2006 vor der Vorlesung

Hinweise: Die Lösungen sollten grundsätzlich schriftlich, Programme zusätzlich auf Diskette oder per E-Mail an Ihren Tutor oder Ihre Tutorin abgegeben werden. Die Abgabe ist in Gruppen bis zu zwei Personen erlaubt.

3. Primzahlenfaktorisierung

3 Punkte

Schreiben Sie eine Funktion `factors :: Integer -> [Integer]`, die die Liste aller Primfaktoren einer gegebenen Zahl bestimmt.

4. Zeichenketten

5 Punkte

(a) Definieren Sie eine Funktion `count :: Char -> String -> Int`, die die Anzahl der Vorkommen eines Buchstabens in einer Zeichenkette bestimmt. Ist Ihre Funktion endrekursiv? / 2

(b) Schreiben Sie eine Funktion `blowup :: String -> String`, die eine Zeichenkette $c_1c_2c_3 \dots$ in die Form $c_1c_2c_2c_3c_3c_3 \dots$ konvertiert. / 3

5. Alternative Implementierung des Picture-Moduls

4 Punkte

In dem in Aufgabe 2 verwendeten Modul `Picture.hs` werden Bilder als Listen von Zeichenketten implementiert, d.h. Bilder werden als Listen von Listen von Zeichen dargestellt:

```
type Picture = [String] -- = [[Char]]
```

In dieser Aufgabe soll eine alternative Bildimplementierung betrachtet werden, die sogenannte *Laufmäßenkodierung*:

```
type Picture = [(Char,Int)]
```

Zu jedem Zeichen wird angegeben, wie oft dieses wiederholt wird. Eine Zeichenkette `"###.##"` wird etwa durch die Liste `[(' ',3),('.',2),('#',1)]` dargestellt.

Definieren Sie die in dem auf der Vorlesungsseite angegebenen Modul `PictureLK.hs` fehlenden Funktionen `above`, `aside`, `flipH`, `flipV` und `invertColour` (ohne eine Umwandlung in die Zeichenkettendarstellung durchzuführen).

Hinweis: Bei einigen Funktionen ist der Einsatz von Listenabstraktionen von Vorteil.