

1.	2.	3.	4.	5.	6.	Σ

Klausur zu ALGORITHMEN UND PROGRAMMIERUNG I
12. Februar 2001

- **Abgabe pünktlich um 14.00 Uhr! Bitte auf jedes Blatt den Namen und die Matrikelnummer schreiben und alle Blätter zusammen mit dem Aufgabenzettel abgeben!**
- Die Klausur ist bestanden, wenn 50 % der Punkte erreicht werden. Es gibt insgesamt 34 Punkte, d.h. zum Bestehen der Klausur müssen 17 Punkte erreicht werden.
- Es dürfen nur die selbst mitgebrachte Unterlagen verwendet werden; Tausch oder Teilen von Unterlagen während der Klausur ist nicht möglich.
- Achten Sie bitte auf die Robustheit Ihrer Definitionen und auf die Angabe der Signaturen.

Vorname:

Nachname:

Matrikelnummer:

1. Aufgabe

4 Punkte

Schreiben Sie (ohne Verwendung der Systemfunktionen *lcm* und *gcd*) eine Haskell-Funktion zur Berechnung des kleinsten, gemeinsamen Vielfachen (*kgV*) zweier natürlicher Zahlen.

2. Aufgabe

8 Punkte

- (a) Zu einer Menge M heißt $P(M) := \{T \mid T \subseteq M\}$ Potenzmenge von M . Zur Erinnerung: die Potenzmenge ist die Menge aller Teilmengen einer gegebenen Menge. Übertragen Sie dieses Konzept auf endliche Listen und definieren Sie einen entsprechenden Operator $pot :: [t] \rightarrow [[t]]$.
Beispiel: $pot [1, 2, 3] = [[], [1], [2], [3], [1, 2], [1, 3], [2, 3], [1, 2, 3]]$.

Hinweis: diese Aufgabe erlaubt mehrere Lösungen, insbesondere ist die Reihenfolge der Elemente in $pot l$ unerheblich.

- (b) Wenden Sie Ihre Funktion pot auf die Liste $[1, 2]$ an und reduzieren Sie Ihren Ausdruck, so dass Sie $[[], [1], [2], [1, 2]]$ bzw. eine Permutation davon erhalten.

bitte wenden

3. **Aufgabe**

6 Punkte

Gegeben sei eine Liste und ein Prädikat, das über Elementen dieser Liste definiert ist. Schreiben Sie je eine Haskell-Funktion, die

- (a) das erste Element liefert, das das Prädikat erfüllt
- (b) alle Elemente liefert, die das Prädikat erfüllen
- (c) die Liste der Indizes liefert, an denen Elemente stehen, die das Prädikat erfüllen.

4. **Aufgabe**

6 Punkte

Gegeben sei ein Text als Liste seiner Wörter. Schreiben Sie eine Haskell-Funktion *sp*, die solch einen Text spaltenweise ausgibt, z.B.:

```
> sp ["das", "ist", "schön"]
```

```
d i s
a s c
s t h
   ö
   n
```

5. **Aufgabe**

4 Punkte

Beweisen Sie durch strukturelle Induktion, dass

$$\text{summe } (l1 ++ l2) = \text{summe } l1 + \text{summe } l2$$

gilt, wobei $\text{summe} = \text{foldr } (+) 0$. Begründen Sie jeden Ihrer Beweisschritte.

6. **Aufgabe**

6 Punkte

Gegeben seien die Funktionen:

$$\text{run} :: [t] \rightarrow u \rightarrow u$$

und

$$\text{execute} :: t \rightarrow u \rightarrow u$$

mit

$$\begin{aligned} \text{run } [] s &= s \\ \text{run } (i : il) s &= \text{run } il (\text{execute } i s) \end{aligned}$$

und *execute* beliebig. Beweisen Sie durch strukturelle Induktion, dass

$$\text{run } (il1 ++ il2) s = \text{run } il2 (\text{run } il1 s)$$

für alle Argumente *s* und *il2* sowie endliche Listen *il1* gilt. Begründen Sie jeden Ihrer Beweisschritte.