

11. Übung zu Informatik I im WS 16/17
Abgabe: 23. Januar 2017, 12⁰⁰ Uhr.

Hinweis: Bitte geben Sie für alle Ihre Funktionen eine entsprechende Signatur an, da das Easy-System Ihre Lösung ansonsten ggf. nicht richtig interpretieren kann! Die zu verwendenden Signaturen entnehmen Sie bitte den Aufgabenstellungen. Achten Sie insbesondere bei Zahlen darauf, dass Sie für Ihre Abgaben den Typ `Int` verwenden und nicht `Integer`, da dies zwei unterschiedliche Datentypen sind.

Aufgabe 26 - Gruppenangabe (10 + 10 = 20 Punkte)
Implementieren Sie ein Telefonbuch in Haskell! Erstellen Sie dafür einen neuen Datentyp `Telefonbuch` auf zwei verschiedene Varianten, welche jeweils die Beziehung von Namen zu Telefonnummern darstellen.

- a) Implementieren Sie den Datentyp als Liste von String Tupeln:
`type Telefonbuch = [(String, String)]`
Der erste Wert eines Tupels stellt dabei den Namen einer Person dar und der zweite Wert dessen Telefonnummer (bspw. „+49 251 / 83 - 0“).
- b) Implementieren Sie den Datentyp als Funktion:
`type Telefonbuch = String -> String`
Die Funktion bildet die Menge von Namen auf die Menge von Telefonnummern ab.

Der Modulname der Datei soll in beiden Fällen `Telefonbuch` sein. Bitte importieren Sie die Datei `Data.hs` mit Hilfe des Befehls `import Data` nach der Moduldeklaration. Die Datei `Data.hs` kann im Learnweb heruntergeladen werden und enthält die jeweilige Definition des Telefonbuchs. Zum Testen müssen Sie die Datei `Data.hs` neben die Datei `Telefonbuch.hs` legen. Implementieren Sie außerdem für beide Teilaufgaben folgende Funktionen:

- Eine Funktion `leeresTelefonbuch :: Telefonbuch`, welche ein leeres Telefonbuch zurückgibt. Achten Sie beim Schreiben dieser Funktion darauf, dass die anderen Funktionen Anforderungen an ein leeres Telefonbuch stellen.
- Eine Funktion `eintragen :: String -> String -> Telefonbuch -> Telefonbuch` zum Einfügen eines neuen Eintrags in das Telefonbuch. Der erste String repräsentiert den Namen einer Person und der zweite String deren Telefonnummer. Hier ist eine explizite Überprüfung auf Duplikate nicht notwendig.

- Eine Funktion `suche :: String -> Telefonbuch -> String` zum Nachschlagen einer Telefonnummer zu einem bestimmten Namen. Sollte der String nicht enthalten sein, soll der String `"nicht gefunden"` ausgegeben werden.
- Eine Funktion `sucheMuster :: String -> Telefonbuch -> Telefonbuch`. Diese gibt für eine partielle Telefonnummer ein gekürztes Telefonbuch aus, wobei es alle Einträge des übergebenen Telefonbuches übernimmt, welche mit der partiellen Telefonnummer starten. Greifen Sie bei der Überprüfung, ob eine Telefonnummer mit einer gewissen Zeichenfolge startet, bitte auf keine vordefinierten Methoden zurück, sondern implementieren Sie eine eigene Hilfsfunktion!

Hinweis: If-Konstrukte und `filter` dürfen verwendet werden!

Aufgabe 27 - Gruppenangabe

(5 + 5 = 10 Punkte)

Gegeben ist eine Baumstruktur:

```
data Baum = Zweig Baum Baum | Blatt Int
```

Verwenden Sie innerhalb eines Moduls `BaumOperation` die Funktion `dc` (Divide & Conquer), wie sie in der Vorlesung vorgestellt wurde. Importieren Sie dazu mit Hilfe des Ausdrucks `import dc` die Datei `DC.hs`, welche Sie neben Ihre Datei `BaumOperation.hs` ablegen müssen. Die Datei `DC.hs` können Sie im Learnweb herunter laden. Sie enthält ebenfalls die oben genannte Definition eines Baumes. Implementieren Sie die vier Funktionen `isSimple`, `solve`, `partition` und `combine`, welche von `dc` benötigt werden, um jeweils folgende Probleme zu lösen:

- Berechnen Sie das Maximum der Werte aller Blätter.
- Berechnen Sie die Höhe des Baumes. Die Höhe eines Blattes ist 0. Die Höhe eines Baumes, welcher mehr als ein Blatt enthält, ist um eins größer als die Höhe des höchsten Teilbaumes. Die Höhe von `Zweig (Zweig (Blatt 1) (Blatt 2)) (Blatt 3)` wäre entsprechend 2.

Verwenden Sie für die Funktionen folgende Signaturen:

- `isSimple :: Baum -> Bool`
- `solve :: Baum -> Int`
- `combine :: [Int] -> Int`
- `partition :: Baum -> [Baum]`

Anmerkung: *Versehen Sie ihre Abgaben bitte mit **Namen**, **Matrikelnummern**, **E-Mail-Adressen** und **Studiengängen** der beteiligten Bearbeiter/innen und laden Sie diese in der entsprechenden Aktivität im Learnweb hoch. Aufgaben, die mit dem*

*Hinweis **Gruppenabgabe** versehen sind, dürfen mit maximal 3 Bearbeiter/innen gelöst werden. Aufgaben mit dem Hinweis **Einzelabgabe** müssen von jedem Studenten und jeder Studentin eigenständig gelöst und abgegeben werden. Viele der Aufgaben werden über das **EASy** (E-Assessment System) System im Learnweb eingereicht und automatisch vorausgewertet. Um Probleme bei der Abgabe wegen Überlastung des EASy-Servers zu vermeiden, würde ich Sie bitten, eine Abgabe auf den letzten Drücker nach Möglichkeit zu vermeiden. Bei Fragen zum Übungsbetrieb wenden Sie sich bitte an ihren/ihre Tutor/in oder an Tobias Reischmann.*