



Bitte bearbeiten Sie die folgenden Aufgaben in Zweiergruppen. Wenn Sie Feedback zu Ihren Lösungen haben möchten, können Sie diese in elektronischer Form per Mail an h.g.esser@gmx.de abgeben (bitte keine Mehrfachabgabe identischer Lösungen). Es gibt keine Bewertung/Benotung.

1. Rekursion, Strings, Dictionaries

[Fortsetzung vom letzten Mal]

c) Schreiben Sie ein Python-Programm, das Zahlen als Strings (im Format „123,456“ mit Dezimalkomma) speichert und diese addieren kann. Sie benötigen also eine Funktion `add(x,y)`, die korrekt rechnet und einen String im richtigen Format zurückgibt. Intern soll das Programm Zahlen als Dictionary implementieren und zu jeder Stelle die Ziffer speichern, also z. B.:

123,456 \rightarrow { 2: 1, 1: 2, 0: 3, -1: 4,
 -2: 5, -3: 6 }

0,001 \rightarrow { -3 : 1 }

(Null-Ziffern werden nicht gespeichert, betrachten Sie nur positive Zahlen.) Achten Sie beim Addieren auf Überträge. Die Umwandlung eines Dictionaries in einen String kann die rechts abgebildete Funktion `DictToStr()` erledigen.

```
def DictToStr(d):  
    ks = d.keys()  
    s = ""  
    for i in range(min(ks), max(ks)+1):  
        if i in ks:  
            s = str(d[i]) + s  
        else:  
            s = "0" + s  
        if i == -1:  
            s = "," + s  
    return s
```

2. Pipelines

- a) In der 5-stufigen RISC-Pipeline kollidieren Befehle, die zur selben Zeit in die Phasen Fetch (F) und Memory (M) wechseln (sollen). Welche Kollisionen dieser Art gibt es in der 6-stufigen CISC-Pipeline?
- b) Warum hat bei MMIX eine JMP-Anweisung keine Write-Back-Phase (W), obwohl das PC-Register verändert wird?
- c) Ein Programm bestehe zu
- 25% aus Speicherzugriffen mit W-Phase
 - 10% aus " " " ohne " "
 - 55% aus arithmetischen Befehlen
 - 10% aus Sprungbefehlen
- und es gelten (wie in der Vorl.) die Ausführungszeiten
- $2t$ für die Phasen F, X, M
 - t für die Phasen D, W
- Berechnen Sie den durchschnittlichen Abstand zwischen der Fertigstellung zweier Befehle
- ohne Pipelining
 - mit Pipelining
- und über deren Quotienten die Beschleunigung durch Pipelining.