

## Midterm Prüfung (Flohr) am 12. Mai 2023, 9:00 – 10:00

### Aufgabe 1 (List Comprehension, Periodensystem) [4 P]

Gegeben ist ein Ausschnitt aus dem Periodensystem der Elemente als Liste von Listen in der Variablen `psystem`. In der zweiten Spalte werden die Erdalkalimetalle geführt.

Erzeugen Sie eine Liste, die alle Erdalkalimetalle beinhaltet und deren Kurzbezeichnung auf den Buchstaben 'a' enden.

```
psystem = ["H", "", ""],  
           ["Li", "Be", ""],  
           ["Na", "Mg", ""],  
           ["K", "Ca", "Sc"],  
           ["Rb", "Sr", "Y" ],  
           ["Cs", "Ba", "La"],  
           ["Fr", "Ra", "Ac"]]
```

### Aufgabe 2 (Rekursion) [4 P]

Ein einfacher Kompressionsalgorithmus reduziert die Anzahl der zu speichernden Zeichen eines Strings. So wird der String "aaaabbbcca" auf "a4b3c2a1" komprimiert, wobei die Ziffer (1-9) nach einem Zeichen für die Häufigkeit des Zeichens steht. Implementieren Sie eine rekursive Funktion `uncompress_rec(s_comp)`, die aus dem komprimierten String die ursprüngliche Zeichenkette wiederherstellt.

### Aufgabe 3 (Caesar-Verschlüsselung) [4 P]

Eine einfache, sehr leicht zu brechende Art der Verschlüsselung ist die sogenannte Caesar-Verschlüsselung, bei der die Buchstaben um eine gewisse Anzahl (hier als Verschiebung bezeichnet) zyklisch verschoben werden. Hat z.B. die Verschiebung den Wert 3, so wird aus

```
'a' --> 'd'  
'b' --> 'e'  
...  
'z' --> 'c'
```

Implementieren Sie die Funktion `caesar(wort, verschiebung)`, die auf die Buchstaben im übergebenen Parameter `wort` die im Parameter `verschiebung` angegebene zyklische Verschiebung anwendet und das so entstandene Geheimtext-Wort als Ergebnis zurückliefert.

Der Einfachheit halber kann man voraussetzen, dass das übergebene Wort nur aus Kleinbuchstaben besteht.

#### Aufgabe 4 (csv-Dateien) [8 P]

Nach der Fusion zweier Versandhäuser müssen die Kundendaten dieser zweier Firmen verschmolzen werden. In den Dateien firmaA.csv und firmaB.csv sind die Kundendaten der beiden Firmen enthalten. Glücklicherweise sind die Daten in beiden Dateien im identischen CSV-Format:

NAME;PLZ

Die Einträge sind zudem alphabetisch nach Namen sortiert.

Die Dateien beginnen jeweils in der ersten Zeile mit Daten.

Aus den Kundendaten der beiden Firmen soll eine nach Namen sortierte Gesamt-Liste durch die Anwendung des Merge-Algorithmus, der im Praktikum vorgestellt wurde, erstellt werden.

Schritte der Implementierung:

- Öffnen Sie beide Dateien mit Hilfe eines Kontextmanagers.
- Speichern Sie die Datensätze der beiden Dateien in Listen zwischen.
- Verschmelzen Sie die Datensätze mit dem Merge-Algorithmus zu einer Gesamt-Liste.

#### Aufgabe 5 (Kontextmanager) [10 P]

Zur einfachen Performancemessung eines Algorithmus kann der Kontextmanager von Python verwendet werden.

MyTimer
- title: str - start: float - end: float
+ MyTimer(title_: str) + __enter__(self): MyTimer + __exit__(self, exc_type, exc_value, traceback): void

Implementieren Sie dazu die Klasse MyTimer gemäß dem oben angegebenen UML Klassendiagramm. Realisieren Sie die beiden Methoden `__enter__()` und `__exit__()`, so dass beim Eintritt des Kontextmanagers (enter) die Zeit gestartet, beim Verlassen (exit) die Zeit gestoppt und die vergangene Zeit (end-start) auf der Konsole ausgegeben wird. Der Parameter title des Konstruktors soll bei der Ausgabe in `__exit__()` als Erstes ausgegeben werden.

Hinweise: Die Parameter der Methode `__exit__()` müssen bei der Implementierung zwar in der Signatur angegeben, jedoch können diese bei der Methodenimplementierung ignoriert werden. Die Funktion `time()` des Moduls `time` liefert die vom Zeitpunkt des Aufrufs vergangenen Sekunden seit dem 01.01.1970 um 00:00:00 Uhr als float Wert zurück.