

Übungsaufgabe 3

Programmierung mit Emacs-Lisp

Hinweis: Eine Einführung zu Lisp finden Sie im entsprechenden Kapitel von Pratt und Zelkowitz.¹ Weiterhin sei auf ein Einführung von Chassell hingewiesen.² Allerdings ist für die Übungsblattlösung nur ein Bruchteil davon für Sie im Moment relevant, nämlich im Wesentlichen nur Kapitel 1, 3, 7 und 11. Für alle Aufgaben gilt: Nicht immer müssen Sie genau eine Funktion schreiben. Manchmal ist es hilfreich, zusätzliche Funktionen zu schreiben, welche einzelne Teilaufgaben erfüllen. Für einen maximalen Lernerfolg, vermeiden sie den `setq` Befehl und verwenden Sie Rekursion an den nötigen Stellen.

Alle Programme sind nur in Emacs-Lisp zu schreiben, nicht in irgendeinem anderem Lisp Dialekt! Testen Sie daher Ihre Programme im Emacs (bspw. Pool)!

Aufgabe 9 Implementieren Sie in einer Datei `binom.el` eine Funktion zur Berechnung der Fakultät einer natürlichen Zahl n , sowie eine darauf aufbauend Funktion zur Bestimmung des Binomialkoeffizienten zweier natürlicher Zahlen a und b . Testen Sie beide Methoden mit jeweils mindestens einem Beispiel.

Für welche a arbeitet Ihr Programm, bei gültigem b , noch korrekt? Was können Sie entsprechend über die Rechengenauigkeit von Emacs Lisp sagen? Geben Sie bitte Ihre Antwort in Kommentarform ab.

Aufgabe 10 Die Caesar-Verschlüsselung ist eine einfach Verschiebechiffre. Alle Buchstaben eines Eingabetextes werden dabei eindeutig auf einen Geheimbuchstaben abgebildet. Das Geheimalphabet entsteht durch die zyklische Verschiebung der geordneten Buchstaben des Eingabealphabetes um eine feste Anzahl an Stellen. Diese Anzahl der Verschiebungen stellt somit den Schlüssel zum kodieren und dekodieren von Eingaben dar.

Schreiben Sie in eine Datei `caesar.el` eine Emacs Lisp-Funktion, welche die Caesar Chiffrierung durchführt. Dabei soll zu einer gegebenen Liste von Zahlen $l_i \in \{0 \dots 25\}$, anstelle von Buchstaben, und einer gegebenen Ganzzahl d die Liste $l'_i = (l_i + d) \bmod 26$ zurückgegeben werden. Das heißt, das zu jedem Element der Liste d hinzuaddiert und die Summe dann modulo 26 gerechnet werden soll. Testen Sie ihr Programm an mindestens einem Beispiel.

¹Pratt, Zelkowitz, *Programmiersprachen*. Prentice-Hall, München, 1997

²Chassell, *An introduction to Programming in Emacs Lisp*, 2009, <http://www.gnu.org/software/emacs/emacs-lisp-intro/>

Aufgabe 11 Implementieren Sie Emacs Lisp-Funktionen in einer Datei `mengen.el`, welche die Mengenoperationen $A \cup B$ (Vereinigung), $A \cap B$ (Durchschnitt) und $A \setminus B$ (Differenz, “A ohne B”) auf zwei gegebene Mengen A und B anwenden. Diese Mengen sollen, genau wie die Ergebnisse, in Listen gespeichert sein. Achten Sie darauf, dass nach Ausführen der Operationen kein Element mehrfach vorkommt. Auf Überprüfung auf mehrfaches Vorkommen in den Eingabemengen können Sie verzichten.

Testen Sie jede Funktion an mindestens einem Beispiel.

Aufgabe 12 Schreiben Sie ein Emacs Lisp-Programm `loesche.el`, welches zunächst eine unsortierte Liste ganzer Zahlen ($L_i \in \mathbb{Z}$, $|L| = n$) aufsteigend sortiert. Weiterhin soll das mittlere Element der sortierten Liste ermittelt und gelöscht werden. Die Mitte einer Liste sei hierbei $\lfloor n/2 \rfloor$. Geben Sie danach die sortierte Liste ohne dieses Element aus. Arbeiten Sie auf einer Kopie der Liste so dass die ursprüngliche Liste nach Aufruf des Programms erhalten bleibt.

Beispielsweise soll bei der Liste (`loesche-mitte '(3 4 -1 3 2 4)`) die Liste `(-1 2 3 4 4)` zurückgeben werden. Testen Sie ihr Programm an drei weiteren gewählten Beispielen, welche aus verschiedenen langen Listen bestehen. Überlegen Sie sich insbesondere wie Sie testen, dass die gegebene Liste nicht verändert wird. Beachten Sie auch das der Index einer Liste stets mit Null beginnt. Nutzen Sie bereits vordefinierte Funktionen, beispielsweise zum Sortieren.

Zusatzaufgabe 3 Implementieren Sie ein Emacs Lisp-Programm `sieb.el`, welches zu einem gegebenem Argument n alle Primzahlen zwischen 2 und n nach der Methode des “Sieb des Erathostenes” bestimmt. Eine genauere Beschreibung des “Sieb des Erathostenes” finden Sie in Aufgabe 4 des ersten Übungsblatts. Testen Sie ihr Programm für mindestens drei verschiedene n .