

Übungsserie 4a

Programmierung mit Common-Lisp

Bitte beachten Sie den Abgabzeitpunkt in Moodle. Serien per E-Mail oder verspätete Abgaben können nicht gewertet werden. Die Textaufgaben bitte per PDF mit den Programmieraufgaben zusammen in ein Zip-Archiv bündeln.

Aufgabe 1 (5 Punkte) Schreiben Sie ein rekursives Common-Lisp Programm `laenge.cl`, welches zu einer Liste von Listen die durchschnittliche Listenlänge bestimmt, *ohne* die bereits in Common-Lisp definierten Funktionen wie `length` oder `list-length` zu verwenden. Beispielsweise soll bei der Liste `(mitlaenge '((1 2 3) (A B) (S ta X 2)))` eine 3 zurückgegeben werden.

Testen Sie ihr Programm an drei gewählten Beispielen, welche aus verschiedenen langen Listen bestehen.

Aufgabe 2 (9 Punkte) Implementieren Sie die Common-Lisp Funktionen in einer Datei `mengen.cl`, welche die Mengenoperationen $A \cup B$ (Vereinigung), $A \cap B$ (Durchschnitt), $A \setminus B$ (Differenz, “A ohne B”) und $\mathcal{P}(A)$ (Potenzmenge von A). Wenden Sie die Funktionen auf zwei gegebene Mengen A und B an, *ohne* die bereits in Common-Lisp definierten Funktionen wie `union`, `intersection` oder `set-difference` zu verwenden. Achten Sie bei der Programmierung auf eine funktionale Umsetzung und verwenden Sie nicht die Funktion `(loop ...)`. 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 von Duplikaten in den Eingabemengen können Sie verzichten.

Überlegen Sie sich für jede Funktion 2 Testfälle und testen Sie daran Ihre Funktionen.

Hinweise: Die Funktionen `(find ...)`¹ und `(delete ...)`² können sehr hilfreich sein. Die Testfälle können auch ohne die Funktionen erstellt werden, falls Sie diese nicht implementiert haben.

Aufgabe 3 (6 Punkte) Das “Sieb des Erathostenes” ist ein Verfahren zur Bestimmung aller Primzahlen zwischen 2 und $n \in \mathbb{N}$, welches ohne Division auskommt. Hierzu werden alle ganzen Zahlen von 2 bis n notiert. Wir nennen diese Folge das Sieb. Sei p der kleinste noch nicht gestrichene Wert im Sieb. p ist mit Sicherheit eine Primzahl. Wir streichen im Sieb alle Vielfachen $j \cdot p \leq n$ mit $j \in \mathbb{N} \setminus \{0\}$. Anschließend ist der nächste kleinste nicht gestrichene Wert q die nächste gesuchte Primzahl. Jetzt verfahren wir für q wie zuvor mit p . Diese Schritte wiederholen wir solange bis wir zu einer Primzahl kommen, für die $q^2 > n$ gilt. Da dann im Sieb keine Produkte $a \cdot b$ mit $a > q$ und $b > q$ stehen können, denn solche Produkte sind wegen $q > \sqrt{n}$ größer als n , und da alle Produkte $a \cdot b$ mit $a \leq q$ und/oder $b \leq q$ bereits gestrichen wurden, enthält das Sieb nur noch die Primzahlen zwischen 2 und n .

¹http://clhs.lisp.se/Body/f_find_.htm

²http://clhs.lisp.se/Body/f_rm_rm.htm

Implementieren Sie ein rekursives Programm `sieb.cl`, welches zu einem gegebenen Argument n alle Primzahlen zwischen 2 und n nach der Methode des “Sieb des Erathostenes” bestimmt. Verwenden Sie dabei keine Schleifen.

Testen Sie ihr Programm für mindestens drei verschiedene n .

Hinweise: Sie dürfen die `mod` Funktion nutzen um auf ein Vielfaches einer Zahl zu prüfen. Nutzen Sie auch arithmetische Funktionen und Vergleichsfunktionen. Es bietet sich an Teilaufgaben in (rekursive) Funktionen auszulagern.