Modellierung und Programmierung 2

Hinweise Abgabe: https://moodle2.uni-leipzig.de/course/view.php?id=3907

Übungsserie 3

Programmierung mit Common Lisp

Aufgabe 9 Implementieren Sie in einer Datei `binom.cl` eine Funktion zur Berechnung der Fakultät einer natürlichen Zahl n, sowie eine darauf aufbauende 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 sagen? Geben Sie bitte Ihre Antwort in Kommentarform ab.

Aufgabe 10 Die Cäsar-Verschlüsselung ist eine einfache Verschiebechiffre. Alle Buchstaben eines Eingabetextes werden dabei eindeutig auf einen Geheimbuchstaben abgebildet. Das Geheimalphabet ensteht 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 ein Programm `caesar.cl`, welches die Cäsar-Chiffrierung durchführt. Dabei soll das Programm alle ASCII-Buchstaben [A-Z] sowie [a-z] mit einem festen Schlüssel verschieben. Alle Großbuchstaben werden wieder auf Großbuchstaben abgebildet und alle Kleinbuchstaben wieder auf Kleinbuchstaben verschoben. Zeichen, welche keine Buchstaben ([A-Za-z]) sind, bleiben hierbei intakt und werden nicht verschoben. Wenn eine Verschiebung über das Ende des Alphabets gehen würde, wird sie am Anfang fortgesetzt (zyklisch). Wählen Sie den Schlüssel 13 als Voreinstellung. Testen Sie auch andere Schlüssel: `0`, `26`, oder `1`. Erweitern Sie Ihr Programm so, dass es Zeichen für Zeichen von der Standardeingabe liest und verschoben wieder ausgibt.

Beispiel: Für den Verschiebeschlüssel `13` und der Zeichenkette: `"Hello World!"` ergibt sich demnach: `"Uryyb Jbeyq!"`. Der Verschiebeschlüssel ist im Programmcode festgelegt. Testen Sie Ihr Programm: `echo "Hello World." | clisp caesar.cl`

Aufgabe 11 Implementieren Sie ein Programm `mengen.cl`, welches die Mengenoperationen $A \cup B$ (Vereinigung), $A \cap B$ (Durchschnitt) und $\mathcal{P}(A)$ (Potenzmenge) auf gegebene Mengen definiert, ***ohne*** die bereits in Common Lisp definierten vorhandenen Funktionen `union` und `intersection` zu verwenden. 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. Hinweis: Die Potenzmenge $\mathcal{P}(A)$ ist die Menge aller möglichen Teilmengen aus A, inklusive der leeren Menge (`NIL`) und der Menge A selbst.

Aufgabe 12 Implementieren Sie ein Programm `sieb.cl` gemäß Aufgabenstellung Serie 1, Aufgabe 4 in Common Lisp.

Zusatzaufgabe 3 Schreiben Sie zunächst ein Programm `quer.cl`, welches die Quersumme einer Zahl n berechnet. Dabei sei n als Integer gegeben, nicht als Liste von Ziffern.

Eine Verallgemeinerung sind gewichtete Quersummen, bei denen die Ziffern erst mit den Werten einer periodischen Zahlenfolge multipliziert und diese Ergebnisse aufsummiert werden. Dabei wird mit der niedrigstwertigen Ziffer begonnen. Hierbei sei die Folge:

$$1, 3, 2, -1, -3, -2, 1, 3, 2, -1, -3, -2, ...$$

und die zu prüfende Zahl 1234567. Dann ist die gewichtete Quersumme:

$$7 \cdot 1 + 6 \cdot 3 + 5 \cdot 2 + 4 \cdot (-1) + 3 \cdot (-3) + 2 \cdot (-2) + 1 \cdot 1 = 7 + 18 + 10 - 4 - 9 - 4 + 1 = 19$$

Erweitern Sie Ihr Programm so, dass es auch die oben gezeigte gewichtete alternierende Quersumme einer Zahl berechnen kann.