Aufgabe 1: Eine Zahl $m \in \mathbb{N}$ ist ein echter Teiler einer Zahl $n \in \mathbb{N}$ genau dann, wenn m ein Teiler von n ist und wenn außerdem $m < n$ gilt.

Eine Zahl $n \in \mathbb{N}$ heißt perfekt, wenn n gleich der Summe aller echten Teiler von n ist. Zum Beispiel ist die Zahl 6 perfekt, denn die Menge der echten Teiler von 6 ist $\{1, 2, 3\}$ und es gilt $1 + 2 + 3 = 6$.

(a) Implementieren Sie eine Prozedur `echteTeiler`, so dass der Aufruf `echteTeiler`(n) für eine natürliche Zahl n die Menge aller echten Teiler von n berechnet.

(b) Berechnen Sie die Menge aller perfekten Zahlen, die kleiner als $10\,000$ sind.

Aufgabe 2:

(a) Implementieren Sie eine Prozedur `gt`, so dass der Aufruf `gt`(m, n) für zwei natürliche Zahlen m und n die Menge aller gemeinsamen Teiler von m und n berechnet.

Hinweis: Berechnen Sie zunächst die Menge der Teiler von m und die Menge der Teiler von n. Überlegen Sie, wie die Mengenlehre Ihnen weiterhilft, wenn Sie diese beiden Mengen berechnet haben.

(b) Implementieren Sie nun eine Prozedur `ggt`, so dass der Aufruf `ggt`(m, n) den größten gemeinsamen Teiler der beiden Zahlen m und n berechnet.

Aufgabe 3: Implementieren Sie eine Prozedur `kgv`, so dass der Aufruf `kgv`(m, n) für zwei natürliche Zahlen m und n das kleinste gemeinsame Vielfache der Zahlen m und n berechnet.

Hinweis: Es gilt `kgv`$(m, n) \leq m \cdot n$.

Aufgabe 4: Bei den folgenden beiden Teilaufgaben sollen Sie den Operator "`**`", mit dem Sie in SetlX die Potenz-Menge einer Menge berechnen können, nicht benutzen.

(a) Implementieren Sie eine Funktion `subsets`, so dass `subsets`(M, k) für eine Menge M und eine natürliche Zahl k die Menge aller der Teilmengen von M berechnet, die genau k Elemente haben.

Hinweis: Versuchen Sie, die Funktion `subsets`(M, k) rekursiv zu implementieren.

(b) Implementieren Sie eine Funktion `power`, so dass `power`(M) für eine gegebene Menge M die Potenz-Menge von M berechnet. Sie sollen diese Funktion rekursiv implementieren und sich dabei an dem Beweis im Skript zur Vorlesung "Lineare Algebra" orientieren, mit dem wir gezeigt haben, dass für eine endliche Menge M die Gleichung

$$\mathtt{card}(2^M) = 2^{\mathtt{card}(M)}$$

gilt.

(c) Geben Sie nun eine alternative Implementierung der Funktion `power` an, bei der Sie die Funktion `subsets` aus Teil (a) dieser Aufgabe verwenden.

Aufgabe 5: Eine Liste der Form $[a, b, c]$ wird als geordnetes pythagoreisches Tripel bezeichnet, wenn

$$a^2 + b^2 = c^2 \quad \text{und} \quad a < b$$

gilt. Beispielsweise ist $[3, 4, 5]$ ein geordnetes pythagoreisches Tripel, denn $3^2 + 4^2 = 5^2$.

(a) Implementieren Sie eine Prozedur `pythagoras`, so dass der Aufruf

`pythagoras`(n)

die Menge aller geordneten pythagoreischen Tripel $[a, b, c]$ berechnet, für die $c \leq n$ ist.

(b) Ein pythagoreisches Tripel $[a, b, c]$ ist ein reduziertes Tripel, wenn die Zahlen a, b und c keinen nicht-trivialen gemeinsamen Teiler haben. Implementieren Sie eine Funktion `isReduced`, die als Argumente drei natürliche Zahlen a, b und c erhält und die genau dann `true` als Ergebnis zurück liefert, wenn das Tripel $[a, b, c]$ reduziert ist.

(c) Implementieren Sie eine Prozedur `reducedPythagoras`, so dass der Aufruf

`reducedPythagoras`(n)

die Menge aller geordneten pythagoreischen Tripel $[a, b, c]$ berechnet, die reduziert sind.

Berechnen Sie mit dieser Prozedur alle reduzierten geordneten pythagoreischen Tripel $[a, b, c]$, für die $c \leq 50$ ist.

Aufgabe 6: Nehmen Sie an, ein Spieler hat im Poker (Texas Hold'em) die beiden Karten $\langle 8, \heartsuit \rangle$ und $\langle 9, \heartsuit \rangle$ erhalten. Schreiben Sie ein SetlX-Programm, dass die folgenden Fragen beantworten.

(a) Wie groß ist die Wahrscheinlichkeit, dass im Flop wenigsten zwei weitere Karten der Farbe $\heartsuit$ liegen?

(b) Wie groß ist die Wahrscheinlichkeit, dass alle drei Karten im Flop die Farbe $\heartsuit$ haben?

Aufgabe 7: Ein Anagramm eines gegebenen Wortes v ist ein Wort w, dass aus dem Wort v durch Umstellung von Buchstaben entsteht. Beispielsweise ist das Wort "`atlas`" ein Anagramm des Wortes "`salat`". Implementieren Sie eine Funktion `anagram`(s), die für ein gegebenes Wort s alle Wörter berechnet, die sich aus dem Wort s durch Umstellung von Buchstaben ergeben. Die Menge dieser Wörter soll dann als Ergebnis zurück gegeben werden. Es ist nicht gefordert, dass die Anagramme sinnvolle Wörter der deutschen Sprache sind. Beispielsweise ist auch das Wort "`talas`" ein Anagramm des Wortes "`salat`".

Aufgabe 8: Nehmen Sie an, dass Sie n Würfel haben, deren Seiten mit den Zahlen 1 bis 6 bedruckt sind. Weiter ist eine feste Zahl s vorgegeben. Entwickeln Sie eine SetlX-Prozedur `numberDiceRolls`, so dass der Aufruf

`numberDiceRolls`(n, s)

die Anzahl der Möglichkeiten berechnet, mit n Würfeln in der Summe die Zahl s zu würfeln. Beispielsweise soll `numberDiceRolls`$(3, 5)$ den Wert 6 liefern, denn es gibt 6 Möglichkeiten, um mit drei Würfeln in der Summe eine 5 zu würfeln:

$$\langle 1, 1, 3 \rangle, \langle 1, 2, 2 \rangle, \langle 1, 3, 1 \rangle, \langle 2, 1, 2 \rangle, \langle 2, 2, 1 \rangle, \langle 3, 1, 1 \rangle$$

Hinweis: Implementieren Sie eine geeignete Hilfsfunktion.