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.

Hinweis: Am einfachsten ist es, die Funktion `power`(M) rekursiv zu implementieren.

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

$$\texttt{pythagoras}(n)$$

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

$$\texttt{reducedPythagoras}(n)$$

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 $\langle 8, \spadesuit \rangle$ und $\langle 9, \spadesuit \rangle$ 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 $\spadesuit$ liegen?

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