ASSIGNMENT #8 SOLUTION

1. Program

```
/* romanToDecimal takes a list of Roman numerals and returns the decimal     */
/* equivalent if the Roman number is well formed and error otherwise.        */

romanToDecimal(RomanNumerals, DecimalNumber) :-
  accumulate(RomanNumerals, 0, DecimalNumber).

/* accumulate traverses the list of Roman numerals, accumulating the decimal */
/* equivalent using its second argument.                                     */

accumulate([], DecimalValue, DecimalValue).

accumulate([NextRomanNumeral], DecimalValue, Result) :-
  map(NextRomanNumeral, NextDecimalValue),
  Result is DecimalValue + NextDecimalValue.

accumulate([NextRomanNumeral, NextNextRomanNumeral | RestOfRomanNumerals],
    DecimalValue, Result) :-
  map(NextRomanNumeral, NextDecimalValue),
  map(NextNextRomanNumeral, NextNextDecimalValue),
  NextDecimalValue < NextNextDecimalValue,
  computePair(NextDecimalValue, NextNextDecimalValue, PairValue),
  AccumulatedValue is DecimalValue + PairValue,
  accumulate(RestOfRomanNumerals, AccumulatedValue, Result).

accumulate([NextRomanNumeral, NextNextRomanNumeral | RestOfRomanNumerals],
    DecimalValue, Result) :-
  map(NextRomanNumeral, NextDecimalValue),
  map(NextNextRomanNumeral, NextNextDecimalValue),
  NextDecimalValue >= NextNextDecimalValue,
  AccumulatedValue is DecimalValue + NextDecimalValue,
  accumulate([NextNextRomanNumeral | RestOfRomanNumerals], AccumulatedValue,
    Result).

/* map converts a single Roman numeral into its decimal equivalent. */

map(i, 1).
map(v, 5).
map(x, 10).
map(l, 50).
map(c, 100).
map(d, 500).
map(m, 1000).

/* computePair takes 2 decimal equivalents of Roman numerals which have been */
/* determined to require subtraction, checks the legality of the pair, and   */
/* performs the subtraction. Subtractions are used instead of writing four   */
/* consecutive occurrences of the same symbol (i.e., IV should be used       */
/* instead of IIII, MCM instead of MDCCCC, etc.). Only symbols which are     */
/* powers of 10 may be used in subtractions (i.e., I, X, C, and M).          */
/* Furthermore, subtraction rules require that a symbol representing 10^x    */
```

```
/* may not precede any symbol larger than 10^(x+1) (e.g., IC is not          */
/* permitted to represent 99 but XC is 90).                                  */

computePair(N1, N2, Result) :-
  power10(N1),
  N1times5 is N1 * 5,
  N1times5 = N2,
  Result is N2 - N1.

computePair(N1, N2, Result) :-
  power10(N1),
  N1times10 is N1 * 10,
  N1times10 = N2,
  Result is N2 - N1.

/* power10 determines whether the argument is a power of 10 or not. */

power10(1).
power10(N) :- N > 1, Ndiv10 is N // 10, power10(Ndiv10).
```

Output

```
| ?- romanToDecimal([c, d, v], N).

N = 405 ?

yes
| ?- romanToDecimal([m, m, v, i, i], N).

N = 2007 ?

yes
| ?- romanToDecimal([m, m, m, c, m, x, c, i, x], N).

N = 3999 ?

yes
| ?- romanToDecimal([m, i, m], N).

no
```

2. Program

Output

3. P: $\{m > 0 \ \& \ n \geq 0\}$ Precondition

S_1 a := m;

S_2 b := n;

S_3 k := 1;

	I: $\{b \geq 0 \ \& \ k \times a^b = m^n\}$	Loop Invariant
S_4	while (b > 0) loop	
	b := b - 1;	
	k := k * a;	
	end loop;	
	Q: $\{k = m^n\}$	Postcondition

Proof

To prove the above program correct, we divide it into two parts at the loop invariant assertion, and then prove each part in turn. This follows from the Composition rule.

$$\frac{\{P\}S_1; S_2; S_3\{I\}, \{I\}S_4\{Q\}}{\{P\}S_1; S_2; S_3; S_4\{Q\}} \tag{1}$$

(a) $\{m > 0 \ \& \ n \geq 0\}$ a := m; b := n; k := 1; $\{b \geq 0 \ \& \ k \times a^b = m^n\}$

We next use the Composition rule.

$$\frac{\{m > 0 \ \& \ n \geq 0\}a := m; b := n\{P_3\}, \{P_3\}k := 1\{b \geq 0 \ \& \ k \times a^b = m^n\}}{\{m > 0 \ \& \ n \geq 0\}a := m; b := n; k := 1; \{b \geq 0 \ \& \ k \times a^b = m^n\}} \tag{2}$$

To derive P_3, we use the axiom of Assignment on the statement k := 1.

$$\{b \geq 0 \ \& \ 1 \times a^b = m^n\}k := 1\{b \geq 0 \ \& \ k \times a^b = m^n\} \tag{3}$$

Therefore, $P_3 \equiv b \geq 0 \ \& \ a^b = m^n$.

We now have to prove $\{m > 0 \ \& \ n \geq 0\}a := m; b := n\{b \geq 0 \ \& \ a^b = m^n\}$. This requires another application of the Composition rule.

$$\frac{\{m > 0 \ \& \ n \geq 0\}a := m\{P_2\}, \{P_2\}b := n\{b \geq 0 \ \& \ a^b = m^n\}}{\{m > 0 \ \& \ n \geq 0\}a := m; b := n\{b \geq 0 \ \& \ a^b = m^n\}} \tag{4}$$

To derive P_2, we use the axiom of Assignment on the statement b := n.

$$\{n \geq 0 \ \& \ a^n = m^n\}b := n\{b \geq 0 \ \& \ a^b = m^n\} \tag{5}$$

Therefore, $P_2 \equiv n \geq 0 \ \& \ a^n = m^n$.

Finally, we must prove $\{m > 0 \ \& \ n \geq 0\}a := m\{b \geq 0 \ \& \ a^n = m^n\}$. First we apply the Assignment axiom to derive the weakest precondition for the assignment, which we call P_1.

$$\{n \geq 0 \ \& \ m^n = m^n\}a := m\{n \geq 0 \ \& \ a^n = m^n\} \tag{6}$$

Therefore, $P_1 \equiv n \geq 0$.

Applying the Consequence-2 rule completes the proof of this part.

$$\frac{m > 0 \ \& \ n \geq 0 \Rightarrow n \geq 0, \{n \geq 0\}a := m\{n \geq 0 \ \& \ a^n = m^n\}}{\{m > 0 \ \& \ n \geq 0\}a := m\{n \geq 0 \ \& \ a^n = m^n\}} \tag{7}$$

(b) $\{b \geq 0 \ \& \ k \times a^b = m^n\}$

```
while (b > 0) loop
    b := b - 1;
    k := k * a;
end loop;
```

$\{k = m^n\}$

To apply the Loop rule, we need the postcondition to be in the form $I \ \& \ \neg(b > 0)$ $\equiv b = 0 \ \& \ k = m^n$. Therefore, we first apply the Consequence-1 rule.

$$\frac{\{I\}S_4\{I \ \& \ \neg(b > 0)\}, I \ \& \ \neg(b > 0) \Rightarrow k = m^n\}}{\{I\}S_4\{k = m^n\}} \tag{8}$$

Now we may apply the Loop rule

$$\frac{\{\mathrm{I}\ \&\ \mathrm{b} > 0\}\mathrm{b} := \mathrm{b} - 1; \mathrm{k} := \mathrm{k} * \mathrm{a}\{\mathrm{I}\}}{\{\mathrm{I}\}\mathrm{S}_4\{\mathrm{I}\ \&\ \neg(\mathrm{b} > 0)\}} \tag{9}$$

followed by the Composition rule.

$$\frac{\{\mathrm{I}\ \&\ \mathrm{b} > 0\}\mathrm{b} := \mathrm{b} - 1\{\mathrm{P}_4\}, \{\mathrm{P}_4\}\mathrm{k} := \mathrm{k} * \mathrm{a}\{\mathrm{I}\}}{\{\mathrm{I}\ \&\ \mathrm{b} > 0\}\mathrm{b} := \mathrm{b} - 1; \mathrm{k} := \mathrm{k} * \mathrm{a}\{\mathrm{I}\}} \tag{10}$$

To determine P_4, we apply the Assignment axiom.

$$\{\mathrm{b} \geq 0\ \&\ \mathrm{k} \times \mathrm{a} \times \mathrm{a}^{\mathrm{b}} = \mathrm{m}^{\mathrm{n}}\}\mathrm{k} := \mathrm{k} * \mathrm{a}\{\mathrm{b} \geq 0\ \&\ \mathrm{k} \times \mathrm{a}^{\mathrm{b}} = \mathrm{m}^{\mathrm{n}}\} \tag{11}$$

Therefore, $\mathrm{P}_4 \equiv \mathrm{b} \geq 0\ \&\ \mathrm{k} \times \mathrm{a}^{\mathrm{b}+1} = \mathrm{m}^{\mathrm{n}}$.
We apply the Assignment axiom again to prove

$$\{\mathrm{b} \geq 0\ \&\ \mathrm{k} \times \mathrm{a}^{\mathrm{b}} = \mathrm{m}^{\mathrm{n}}\}\mathrm{b} := \mathrm{b} - 1\{\mathrm{b} \geq 0\ \&\ \mathrm{k} \times \mathrm{a}^{\mathrm{b}+1} = \mathrm{m}^{\mathrm{n}}\} \tag{12}$$

This completes the proof.

Thus the partial correctness proof of the given program consists of steps 1-12.