CS 221 Functional Programming I

Coursework 2

Question 1. Modular exponentiation is defined by

\[ \text{expmod}(b, e, m) = b^e \mod m \]

where \( b, e, m \) are integers with \( b > 0, e \geq 0 \) and \( m > 0 \). Modular exponentiation plays an important role in cryptography.

Give an efficient recursive definition of \( \text{expmod} \) in Haskell using the following idea of repeated squaring: Assume \( e > 0 \) and set

\[ f = e \div 2 \]
\[ p = \text{expmod}(b, f, m) \]
\[ c = b \mod m \]

Then one can express \( \text{expmod}(b, e, m) \) in terms of \( c, p^2 \) and \( m \) using multiplication and \( \mod \).

In order to find the solution you should distinguish the cases when \( e \) is even or odd, and you may use the fact that \( \mod \) commutes with multiplication, that is,

\[ (x_1 \ldots x_k) \mod m = ((x_1 \mod m) \ldots (x_k \mod m)) \mod m \]

Use your fast implementation of \( \text{expmod} \) to compute the last 6 digits (in decimal notation) of \( x^2 \) where \( x \) is your student number. [20 marks]

Question 2. You are at city \( n \) where \( n \) is your student number. Your goal is to reach city 1. The rules for the journey are as follows:

If you are at city \( k \), then

- if \( k \) is even, go to city \( k/2 \),
- if \( k \) is odd, go to city \( 3 \times k + 1 \).

Compute the list of all cities visited on your journey. What is the largest city (number) visited? [20 marks]
**Question 3.** Newton’s method:

- To approximate $\sqrt{x}$ start with 1 (or any other value) as first approximation.
- If $y$ is an approximation of $\sqrt{x}$, then $(y + x/y)/2$ is a better approximation.
- Stop if the approximation $y$ is good enough, say $|y^2 - x| < 0.00001$.

Hints: Use the predefined functional until. Beware of rounding errors!

Implement a variant of the Newton method where the stopping condition is given by an upper bound for the number of iterations, and the result is the list of all approximations computed. [20 marks]

In the following we mean by a point an object of type Pt (= (Float,Float)).

**Question 4.** Reflect a given list of points at the x-axis. [15 marks]

**Question 5.** Compute the corners of a regular polygon with $n$ corners, centered at the origin, and with one corner at point $(1, 0)$. [15 marks]

**Question 6.** Create the $n \times n$ grid, that is the list of all points $(x, y)$ with $-n \leq x, y \leq n$ and $x, y$ ”integers”. [15 marks]

**Question 7.** Compute the list of all points with integer coordinates that lie within the circle of a given radius around a given point. [15 marks]

**Due date:** 8 November 2004

**Notes:**
1. Use the template available at [http://www.cs.swan.ac.uk/~csulrich/cs221html](http://www.cs.swan.ac.uk/~csulrich/cs221html). It contains all functions from the Coursework 1 that might be useful.
2. Otherwise the same conditions as for Coursework 1 apply.