Introduction to Programming 1: Assignment 1

Due: September 14, 2017. 11 pm

Important Instructions: Submit your solution in a single file named **`loginid.1.hs`** on Moodle. For example, if I were to submit a solution, the file would be called **`spsuresh.1.hs`**. You may define auxiliary functions in the same file, but the solutions should have the function names specified by the problems.

---

1. Define a function `isPrime :: Integer -> Bool` that checks if the given input (a positive integer) is a *prime number*.

   Sample cases:

   ```
   isPrime 1          = False
   isPrime 2          = True
   isPrime 9831655609 = True
   isPrime 8128       = False
   ```

2. The *Goldbach conjecture* states that every even integer greater than 2 can be expressed as the sum of two primes. Define a function

   ```
   goldbachPartition :: Integer -> (Integer, Integer)
   ```

   such that `goldbachPartition a` returns `(b, c)` where b and c are primes, $b \leq c$, and $a = b + c$. Assume that the input is always an even integer greater than 2.

   Sample cases:

   ```
   goldbachPartition 8128    = (5, 8123)
   goldbachPartition 8128910 = (103, 8128807)
   ```

3. Define a function `isPerfect :: Integer -> Bool` that checks if the given input (a positive integer) is a *perfect number*. A positive integer is perfect if it is the sum of all its proper divisors.

4. Define a function `nextPerfect :: Integer -> Integer` such that for each positive integer n, `nextPerfect n` returns the least perfect number $m > n$.

5. Define a function `connected :: [(Int,Int)] -> Bool` that checks whether the input list of pairs of integers is *connected*. A list of pairs is connected iff the first component of each pair (other than the very first pair) is the successor of the second component of the previous pair. Sample cases:

   ```
   connected [] = True
   connected [0,1)] = True
   connected [(0,1), (2,15), (16,22), (23,10)] = True
   connected [(0,1), (2,15), (30,22), (23,10)] = False
   ```

6. The *Collatz function* c is defined for positive integers as follows:

$$c(n) = \begin{cases} \frac{n}{2} & \text{if } n \text{ is even} \\ 3n + 1 & \text{otherwise} \end{cases}$$

The *Collatz conjecture* asserts that for all positive n, there exists a nonnegative m such that $c^m(n) = 1$. Define a function `collatz :: Int -> [Int]` which returns the *finite list* of all integers

$$\{c^m(n) \mid m \geq 0, \neg\exists k < m : (c^k(n) = 1)\},$$

if n is positive.

Sample cases:

```
collatz 1    = [1]
collatz 4    = [4,2,1]
collatz (-5) = []
collatz 0    = []
collatz 5    = [5,16,8,4,2,1]
collatz 22   = [22,11,34,17,52,26,13,40,20,10,5,16,8,4,2,1]
```

7. Define `allCycles :: [a] -> [[a]]`, which produces all the cyclic permutations of a given list. You may want to use the Prelude function `cycle` as part of your definition.

Sample cases:

```
allCycles [1,2,3]   = [[1,2,3], [2,3,1], [3,1,2]]
allCycles [1,2,3,4] = [[1,2,3,4], [2,3,4,1], [3,4,1,2], [4,1,2,3]]
```