Haskell Exercises

CMSC 331 Fall 2016

Prepare a Haskell program that includes the following function definitions and PutStrLn calls to show that they work as intended. The program needs to run on the GL platforms. As before, use the UNIX script command to list the program and the results. Email the file to me and the TA Sriraj no later than midnight Tuesday/Wednesday of the last day of class, December 13. Work submitted before by midnight Thursday of the previous week will be eligible for extra credit.

1. Define a function myElem that decides if a value is an element of a list. (Note that elem is already a built-in function in Haskell, but don't use that.) The signature would be:

```
myElem :: Eq a => a -> [a] -> Bool
```

2. Using a high order function, define a function `myReplicate :: Int -> a -> [a]` that produces a list of identical elements. For example,

```
> myReplicate 3 True
[True, True, True]
```

3. Define a function `halve :: [a] → ([ a], [a])` that splits any list into two halves. For example:

```
> halve [1,2,3,4,5,6]
([1, 2, 3], [4, 5, 6])
> halve [1, 2, 3]
([1, 2], [3])
> halve []
([],[])
```

4. A triple (x, y, z) of positive integers is *pythagorean* if x^2 + y^2 = z^2. Using a list comprehension, define a function `pyths :: Int → [( Int, Int, Int)]` that returns the list of all pythagorean triples whose components are at most a given limit. For example:

```
> pyths 10
[(3,4,5),(4,3,5),(6,8,10),(8,6,10)]
```

5. Define a recursive function `merge :: Ord a ⇒ [a] → [a] → [a]` that merges two sorted lists to give a single sorted list. Your definition should be defined using explicit recursion. For example,

```
> merge [2,5,6] [1,3,4]
[1,2,3,4,5,6]
```

6. Using the merge and halve functions from other exercises, define a recursive function `msort :: Ord a ⇒ [a] → [a]` that implements the mergesort algorithm. In mergesort, the empty list and singleton lists are already sorted, and any other list is sorted by merging together the two lists that result from applying mergesort to the two halves of the list separately.