CSCE 330 Fall 2017

Quiz 6

Assigned Thursday, 17-11-28

1. Use the five-step process of Section 6.6 [H] to define a Haskell `sum` function, which computes the sum of a list of `Int`.

   (a) Step1: define the type
   ```
   sum :: [Int] -> Int
   ```

   (b) Step 2: enumerate the cases
   ```
   sum [] =
   sum(x:xs) =
   ```

   (c) Step 3: define the simple cases
   ```
   sum[] = 0
   sum(x:xs) =
   ```

   (d) Step 4: define the other cases
   ```
   sum [] = 0
   sum(x:xs) = x + sum xs
   ```

   (e) Step 5: generalize and simplify
   ```
   sum: Num a => [a] -> a
   sum = foldr (+) 0
   ```
   (not required)

2. Use the five-step process of Section 6.6[H] to define a Haskell function `last`, which selects the last element of a non-empty list.

   (a) Step1: define the type
   ```
   sum :: [a] -> a
   ```

   (b) Step 2: enumerate the cases. (Note: the function is not defined for empty lists.)
   ```
   last(x:xs) =
   ```

   (c) Step 3: define the simple cases

   ```
   last(x:xs) | null xs = x
                  | otherwise =
   ```

   (d) Step 4: define the other cases

   ```
   last(x:xs) | null xs = x
                  | otherwise = last xs
   ```

   (e) Step 5: generalize and simplify (not required)

```
last :: [a] -> a
last [x] = x
last (_:xs) = last xs
```