FUNCTIONAL PROGRAMMING MIDTERM EXAM

90 minutes

April 8, 2014

Id	Full Name	Signature

Q 1	Q 2	Q 3	Total
/ 35	/ 40	/ 25	/ 100

1. What are the results of the following expressions? Briefly explain your answer.

   (a) `[x + y | x <- [1 .. 3], y <- [4 .. 6]]`

   (b) `[x + y | (x, y) <- zip [1 .. 3] [4 .. 6]]`

   (c) `concat (zipWith replicate [3, 4] ['a', 'b'])`

   (d) `map (*2) (filter (>5) [1 .. 7])`

   (e) `(filter (>5) . map (*2)) [1 .. 7]`

   (f) `foldr (\x y -> 2 * y) 1 [1 .. 3]`

   (g) `foldl (\x y -> 2 * y) 1 [1 .. 3]`

2. We want to a write function to find out how many elements in a list are greater than the average of the list.

   (a) Give the type and definition of a function `howManyAbove` that takes a threshold (float) and a list of values (float) and returns the number of elements in the list which are greater than the threshold. Use primitive recursion.

(b) Using the function you have written in (a), give the type and definition of a function `howManyAboveAverage` that takes a list of values and returns the number of elements in the list that are greater the average of the list. (You can use the standard list functions `sum` and `length`.)

(c) Give a tail recursive definition of the `howManyAbove` function.

(d) Give the definition of the `howManyAbove` function as a composition of the `filter` and `length` functions.

(e) Give the definition of the `howManyAbove` function using a `fold` function.

3. Consider the functions given below:

```
frac :: Float -> Float
frac x = x - fromIntegral (floor x)

foo :: Float -> Float
foo x
  | frac (sqrt x) < 0.1 = sqrt x
  | otherwise           = log x
```

$$attempt(f, g, p, x) = \begin{cases} f(x) & if\ p(f(x)) \\ g(x) & if\ \neg p(f(x)) \end{cases}$$

(a) Give the type and definition of the function `attempt` defined as given above.

(b) Give the definition of the function `foo` in terms of the function `attempt` by filling in the blanks below:

```
foo = attempt _____ _____ _____
```