1. Consider a function separate s that returns a 2-tuple with the digits and non-digits in the string s separated, with the initial order maintained.

> separate "July 4, 1776" ("41776","July , ")

> separate "Problem 7: (10 points)" ("710","Problem : ( points)")

Here is a partial implementation of separate, using foldr: separate s = foldr f ([],[]) s

1. Implement map in terms of a fold.
2. Write a function rotabc that changes a's to b's, b's to c's and c's to a's in a string. Only lowercase letters are affected. Using higher order functions.
3. Create a function named dropUntil. This will be a polymorphic higher order function. It’s first parameter, f, will a function that takes one parameter of any type, a, that returns a Boolean result. It’s second parameter, xs, will be a list of values, all of which have type a. The dropUntil function returns a list of values of type a.
4. The following function is supposed to redact some text by replacing all of the characters in a string, other than spaces, with \* characters. Fill in the blank with a lambda function that causes the redact function to perform this task. Do not write a named helper function. redact :: String -> String redact s = map (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) s
5. The following function is supposed to transform every value in a list so that it is equal to the square root of its absolute value. Fill in the blank with a composition of two or more functions that will cause the transform function to perform its intended purpose. Hint: Haskell’s Prelude module includes functions sqrt :: Float -> Float and abs :: Float -> Float. transform :: [Float] -> [Float] transform values = map (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) values