1. (Same as HW2 problem 5.) Write a function

   \texttt{showArea :: IO}

that (i) opens a 600x500 window; (ii) accepts 4 left-button clicks in the window describing a quadrilateral, and outlines this quadrilateral in blue; (iii) calculates the area of the quadrilateral; (iv) draws a red square of equivalent area centered at the origin (of Shape space); (v) waits for a space key to be pressed before closing the window. You can assume the user only enters quadrilaterals that are not self-crossing.

Import and use the \texttt{Shape} and \texttt{Draw} code from the Chs. 2 and 4 of the book to handle the area calculation and the generation of the result square. The graphics function

   \texttt{getLBP :: Window -> IO Point}

waits for a left button click and returns the corresponding mouse location in pixel coordinates. You’ll need to write an inverse translation function from \texttt{Point} to \texttt{Vertex}.

2. Write definitions for the \texttt{map} and \texttt{filter} functions in terms of list comprehensions. (Call them \texttt{map'} and \texttt{filter'} to avoid confusion with the existing Prelude functions.)

3.a. Suppose

   \[ h \ f \ g \ xs = \text{map} \ f \ (\text{map} \ g \ xs) \]

Give an alternative definition for \texttt{h} that builds no intermediate list.

b. Suppose

   \[ r \ p \ q \ xs = \text{filter} \ p \ (\text{filter} \ q \ xs) \]

Give an alternative definition for \texttt{r} that builds no intermediate list.

4. Suppose

   \[ \text{map2} \ f = \text{map} \ (\text{map} \ f) \]

Use \texttt{map2} to define a function

   \[ \texttt{upall :: [String] -> [String]} \]
that converts every String in its argument list to upper-case. (Hint: Check the Prelude for useful functions on characters.)

5. Write a function

   _isPalindrome_ :: String -> Bool

that returns _True_ iff its argument is a palindrome (i.e., a string that reads the same forwards as backwards). Your test should be insensitive to punctuation and alphabetic case, so that, e.g., the classic example “Madam, I’m Adam” will pass. Don’t use any explicit recursion.