COMP0002 Haskell
Lab exercise sheet 4

1. Give definitions of functions to take a list of integers, `ns`, and

   - return the list consisting of all the squares of the integers in `ns`;
   - return the sum of the squares of items in `ns`;
   - check whether all items in the list are greater than zero.

2. Using functions already defined wherever possible write definitions of functions to

   - give the minimum value of a function `f` on inputs `0` to `n`;
   - test whether the values of `f` on inputs `0` to `n` are all equal;
   - test if the values of `f` on inputs `0` to `n` are greater than zero, and
   - check whether the values `f 0`, `f 1` to `f n` are in increasing order.

3. Recall the higher order function `twice` where `twice f x = f(f(x)) = (f . f) x`

   Give the type of twice.

4. Give the type and the definition of a function `iter` so that

   `iter n f x = f (f (f ... (f x) ...))`

   where `f` occurs `n` times in the right hand side of the equation

   For instance

   `iter 3 f x = f (f (f x))`

   and `iter 0 f x` should return `x`

5. Using `iter` and `double` define a function that on input `n` returns 2^n, where n is a whole number and `double` is the integer doubling function from the introductory lecture on Haskell functions.

6. Model the patient's blood type. *ABO Blood Type* refers to `A, B, AB, O` and the positive and negative part refers to *Rhesus (Rh) group*.

   - `Define the type RhType`

- Defining the type ABOType
- Combine ABOType and RhType to create BloodType
- Create BloodType data for 5 patients
- Display your types: showRh, showABO, showBloodType
- Define the canDonateTo function

A can donate to A and AB. B can donate to B and AB. AB can donate only to AB. O can donate to anybody.

7. Use the user defined data type Answer as given in lectures. Define a function called wonky :: Answer -> Answer that changes each answer into a different one from the original (i.e. always into a ‘‘wrong’’ answer). What is the shortest number of times you can apply wonky and get the correct answer using your version of it? What’s the longest? Will that be the same for every version?

8. Use the user defined datatype Shape as used in lectures but replace Circle witha new shape called Ellipse and define a function circle using the Ellipse shape. Redefine the function area :: Shape -> Float replacing the definition for Circle with a new definition for Ellipse.