Haskell Starting Out Questions

A. Sections 2.1–2.3

1. There is an annoyance with negative numbers that is mentioned.
    - What is the correct way to write five times negative three in Haskell?
    - What is a wrong way to do the same thing?
2. Logical operations are basic building blocks of programs. How do you write:
    - and
    - or
    - not
    - equals
    - not equals
3. Haskell follows mostly the normal order of operations from mathematics. However, in math functions must be written `f(x)` with parentheses, whereas in Haskell functions can be written `f x` without parentheses. This means we need to fit functions into the order of operations.
    - Where does function calling fit in the order of operations?
    - Give an example of a problem that this could create. (That is, the program does not do what a math student would expect it to do.)
4. Write the function `hypSquare a b` that computes the square of the length of the hypotenuse of a right triangle with legs `a` and `b`.
5. The apostrophe character (`'`) is usually used for one of two purposes. List both:
    1. Purpose 1:
    2. Purpose 2:
6. Are you allowed to use the apostrophe in function or variable names?
7. In Racket you used (cons x listy) to make a list with x before an existing list. How do you do that in Haskell?
8. In Racket you used (concatenate firstStr secondStr) to put two strings together. How do you do that in Haskell?
9. Slightly longer exercise:
    1. Make a list `learn` containing the words “this” “is” “all” “good”.
    2. Make a new list called `learnMore` containing “Hopefully” “this” “is” “all” “good”
    3. What is `learnMore !! 5`.
10. Complete the table of equivalents

Racket	Haskell language equivalent functions
first	...
rest	...
length	...
empty?	...

11. What are the signatures and purposes of these functions?

    1. init
    2. last
    3. take
    4. drop

12. How can you ask the computer if the list called `xs` contains the number 5?

13. There are several more functions that are used less often, but still make it into the basic starting out chapter. Write quick summaries as necessary.

    1. reverse
    2. maximum
    3. minimum
    4. sum
    5. product

B. Sections 2.4–2.6

1. Create a list of integers from 1 to 1000 (inclusive).

2. Create a list of letters from `'m'` to `'a'` in that order.

3. New functions: write signature and purpose for each.

    1. cycle
    2. repeat
    3. replicate

4. Write a function that removes all of the vowels from a word.

5. What is the main difference between a tuple and a list?

6. New functions: write signature and purpose for each.

    1. fst
    2. snd
    3. zip

7. Write a function that takes in a list of words and puts out a list of ordered pairs (word, length of word).

8. Write a function that takes in a list of words and puts out a list with the second word doubled in its own list, the third word tripled in its own list, etc.

For example, the function call

```
amplify ["hill","rock","star","four"]
```

should produce

```
["hill",["rock","rock"],["star","star","star"], ["four","four","four","four"]]
```