HW 16 **Prof. Caldwell**
Due: 1 November 2011 **COSC 3015**

Exercise 0.1. Read chapter 9 of LYAHFGG. Read Bird pp. 319-322. Real World Haskell also has some good stuff on I/O (see `http://book.realworldhaskell.org/read/io.html`).

Exercise 0.2. In this homework you will do some interactive IO. You are to write an interactive simulator for a queue of numbers.

You can represent the queue itself using a list. When an item is queued up add it to the right end of th list. When an item is dequeued, just take it off the left end of the list. The commands your program should respond to are :

"?"	When this command is entered, display a message describing the commands.
k	If k is a number, queue it up.
"dequeue"	remove an item from the head of the queue.
"quit"	exit the interactive loop.
""	exit the interactive loop on the empty string.
_	For any other input suggest the user try the "?" command and to try again.

Here's the basic structure of the program.

```
import System.IO
import Data.Char
main = chat []
chat xs =
    do putStrLn $ "queue so far = " ++ (show xs)
       putStr "> "
       hFlush stdout
       i ← getLine
       case i of
         "?" → do putStrLn $
                    "commands are: \"dequeue\", \"quit\" or, to queue up a number, enter it."
                  chat xs
         "dequeue" → ...
         "quit" → return ()
         "" → return ()
         _ → if (all isDigit i) then
                ...
             else
                ...
```

Exercise 0.3. For extra credit. Add "save" and "load" commands that allow the current state of the queue to be saved to a file and restored from a file.