HW 7 **Prof. Caldwell**
Due: 25 September 2007 **COSC 3015**

Here is the code given by Hudak in his lecture slides for implementing the Unix wordcount (wc) command.

```
wcf :: (Int,Int,Int) -> String -> (Int,Int,Int)
wcf (cc,w,lc) []          = (cc,w,lc)
wcf (cc,w,lc) (' '  : xs) = wcf (cc+1,w+1,lc) xs
wcf (cc,w,lc) ('\t' : xs) = wcf (cc+1,w+1,lc) xs
wcf (cc,w,lc) ('\n' : xs) = wcf (cc+1,w+1,lc+1) xs
wcf (cc,w,lc) (x : xs)    = wcf (cc+1,w,lc) xs

wc :: IO ()
wc = do name     <- getLine
        contents <- readFile name
        let (cc,w,lc) = wcf (0,0,0) contents
        putStrLn (The file:  ++ name ++ has )
        putStrLn (show cc ++  characters )
        putStrLn (show w  ++  words )
        putStrLn (show lc ++  lines )
```

Not that in this implementation, a file containing only two spaces (or a space and a tab, or a tab and a space or two tabs) will be counted as having two words. This is a bug. Redesign the code so that whitespace is grouped and, while individual spaces are counted as characters, they only are counted as delimiting words if they separate non-word or line delimiting characters.

Provide test cases for you code to show that it does the "right thing." What about a file containing a single space character, or a space followed by a newline?