

## Math 6833 assignments

19. Write the following functions in Haskell. For the first two, try to write them in two or three different ways, making use of different Haskell devices such as guards and `where`.
  1. `threeEqual :: Int -> Int -> Int -> Bool` for which `threeEqual a b c` is `True` if `a`, `b`, and `c` are equal, and `False` otherwise.
  2. `threeDifferent :: Int -> Int -> Int -> Bool` for which `threeEqual a b c` is `True` if no two of `a`, `b`, and `c` are equal, and `False` otherwise.
  3. `roots :: Float -> Float -> Float -> [Float]` where `roots a b c` is a list (possibly the empty list) of the real roots of  $ax^2+bx+cx$ . You can use the built-in `sqrt` function. For simplicity, you may assume that  $a \neq 0$ .
20. `sum :: [Int] -> Int` is the function that sums the entries of a list. Use `map` and `sum` to define the function `length :: [a] -> Int`.
21. Tell what the following Haskell functions calculate:
  1. `func :: Int -> [Int]` defined by  
`func n = [ m | m <- [1..(abs n)], (abs n) `mod` m == 0 ]`
  2. `mystery :: Int -> Bool` defined by  
`mystery n = func n == [1,n]`
  3. `strange :: [a] -> Int` defined by  
`strange list = length ( filter ( == head list ) list )`  
(Note: if you put this in a script, the type must be entered as  
`strange :: Eq a => [a] -> Int` to indicate that the type `a` must have an `==` function.)