2. Give a first order definition for each of the following Haskell functions.

   (a) `nonspace :: [Char] -> Int` which returns the number of non space characters in any given finite list of integers. For example, `nonspace "fun"` evaluates to `3` , and `nonspace "fun prog"` evaluates to `7`. [6]

   (b) `isort :: Ord a => [a] -> [a]` which sorts a given list (over an ordered type) into ascending order using the well known *insertion sort* algorithm. This algorithm is tail recursive, finding the rightful place for the head of a non-empty list in the recursively sorted tail. For example, `isort [4,3,1]` evaluates to `[1,3,4]`. [8]

   (c) `sum :: [Int] -> Int` which sums all of the integers in a given finite list using an **iterative** algorithm. For example, `sum [6,3,7]` evaluates to `16`. [6]

Note that each of the three functions in Question 2 can be defined without reference to the other two.