4. Define each of the following functions in Haskell using a first order function definition and an **iterative** approach.

   (a) `match :: [Char] -> Bool` is a function to parse simple expressions of the following form. Each expression may contain the characters `'('`, `')'`, and a lower case letter in the range `'a'` to `'z'`. For an expression `e` to be correct, that is, for `match e` to evaluate to `True` , the brackets must *match up* in the usual way. For example, `match ""` , `match "(x)"` , and `match "(x(y))"` each evaluates to `True` . And, `match ")("` , `match "(()"` , and `match "(+)"` each evaluates to `False` . [10]

   (b) `final :: [a] -> a` is a function which for the empty list `[]` returns a suitable error message, and for a non-empty list returns the final item. For example, `final "CS256"` evaluates to `'6'` , and `final [1,3,6]` evaluates to `6`. [10]