COMP101P/A Lab 4

1. Write properties that allow you to use QuickCheck to test the following functions from LabSheet1:

   - `halfEvens`
   - `inRange`
   - `countPositives`

2. Write properties that allow you to use QuickCheck to test the following functions from LabSheet2:

   - `rotor`
   - `encipher`
   - `normalise`

   If all else fails you can implement the functions in two different ways then the property can equate them.

3. Consider the Reverse Polish Notation evaluator discussed in lectures. This uses the List function library and the evaluation function is defined as

```
evalRPN :: (Num a, Read a) => String -> a
evalRPN = head . fold procStack [ ] . words

procStack :: (Num a, Read a) => [a] -> String -> [a]
procStack (x : y : ys) "*" = (y * x) : ys
procStack (x : y : ys) "+" = (y + x) : ys
procStack (x : y : ys) "-" = (y - x) : ys
procStack xs numString = read numString : xs
```

   Write your own versions of the functions words and unwords, where

```
words :: String -> [String]
unwords :: [String] -> String
```

   and

```
words "Tomorrow is free" = ["Tomorrow", "is", "free"]
unwords ["boo", "unboo", "not", "frightened"] = "boo unboo not frightened"
```

4. Write properties to test the `words` and `unwords` functions using QuickCheck.
5. Write a function called `standard2RPN` that takes an arithmetic expression and converts it into a string in Reverse Polish Notation.
6. use `standard2RPN` to write a property called `prop_evalRPN` that QuickCheck can use to test `evalRPN`.