4.4

upperNumbers :: [String]

upperNumbers = ["Zero","One","Two","Three","Four","Five","Six","Seven","Eight","Nine","Ten"]

lowerNumbers :: [String]

lowerNumbers = ["zero","one","two","three","four","five","six","seven","eight","nine","ten"]

plural :: Int -> String

plural 0 = ""

plural 1 = " man"

plural n = " men"

line1 :: Int -> String

line1 0 = ""

line1 n = (upperNumbers !! n) ++ (plural n) ++ rest

where rest = " went to mow"

restOfLine3 :: Int -> String

restOfLine3 0 = ""

restOfLine3 1 = "one man and his dog"

restOfLine3 n = (lowerNumbers !! n) ++ plural n ++ ", " ++ restOfLine3 (n-1)

line3 :: Int -> String

line3 0 = ""

line3 1 = "One man and his dog"

line3 n = (upperNumbers !! n) ++ (plural n) ++ ", " ++ restOfLine3 (n-1)

verse :: Int -> String

verse 0 = ""

verse n = (line1 n) ++ "\n" ++ line ++ "\n" ++ (line3 n) ++ "\n" ++ line ++ "\n"

where line = "Went to mow a meadow"

song :: Int -> String

song 0 = ""

song n = song (n-1) ++ "\n" ++ verse n

GABRIEL MOISE – SHEET 2