1 Problem

The n^{th} term of the sequence of triangle numbers is given by, $t_n = \frac{1}{2}n(n+1)$; so the first ten triangle numbers are:

1, 3, 6, 10, 15, 21, 28, 36, 45, 55, ...

By converting each letter in a word to a number corresponding to its alphabetical position and adding these values we form a word value. For example, the word value for SKY is $19+11+25 = 55 = t_{10}$. If the word value is a triangle number then we shall call the word a triangle word.

Using words.txt (right click and 'Save Link/Target As...'), a 16K text file containing nearly two-thousand common English words, how many are triangle words?

2 Solution

```
import Data.List
import qualified Data.Map as Map
import Data.Maybe
import System.Environment
import Data.Char

triNumbers = map (λz → round (0.5 * z * (z + 1))) [1 ..]

split :: Char → String → [String]
split = unfoldr ∘ split'

split' :: Char → String → Maybe (String, String)
split' c l
  | null l = Nothing
  | otherwise = Just (h, drop 1 t)
  where (h, t) = span (≢ c) l

getWords :: String → IO [String]
getWords fname = do
  rawRead ← readFile fname
  let quotedWords = split ',' rawRead
  let dequotedWords = map (λz → (tail ∘ init) z) quotedWords
  return dequotedWords

charToDigit :: Char → Int
charToDigit c = (ord c) − 64

main = do
  let tris = take 30 $ map fromIntegral triNumbers
  myWords ← getWords "words.txt"
  let sums = map (λz → sum (map charToDigit z)) myWords
      tsums = filter (λz → z ∈ tris) sums
      ntris = length tsums
  putStrLn $ "There are a total of " ++ show ntris ++
    " triangle words in the given list"
```

3 Result

```
runhaskell problem42.lhs
There are a total of 162 triangle words in the given list
```