1 Problem

Each character on a computer is assigned a unique code and the preferred standard is ASCII (American Standard Code for Information Interchange). For example, uppercase A = 65, asterisk (*) = 42, and lowercase k = 107.

A modern encryption method is to take a text file, convert the bytes to ASCII, then XOR each byte with a given value, taken from a secret key. The advantage with the XOR function is that using the same encryption key on the cipher text, restores the plain text; for example, $65 XOR 42 = 107$, then $107 XOR 42 = 65$.

For unbreakable encryption, the key is the same length as the plain text message, and the key is made up of random bytes. The user would keep the encrypted message and the encryption key in different locations, and without both "halves", it is impossible to decrypt the message.

Unfortunately, this method is impractical for most users, so the modified method is to use a password as a key. If the password is shorter than the message, which is likely, the key is repeated cyclically throughout the message. The balance for this method is using a sufficiently long password key for security, but short enough to be memorable.

Your task has been made easy, as the encryption key consists of three lower case characters. Using cipher1.txt (right click and 'Save Link/Target As...'), a file containing the encrypted ASCII codes, and the knowledge that the plain text must contain common English words, decrypt the message and find the sum of the ASCII values in the original text.

2 Solution

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

readEncrypted :: String → IO [Integer]
readEncrypted fname = do
  rawRead ← readFile fname
  return $ readIntList rawRead

readIntList [] = []
readIntList xs = (nint : (readIntList xs'))
  where nums = takeWhile isDigit xs
    nint = (read nums :: Integer)
    xs' = dropWhile (λz → (isPunctuation z) ∨ (isControl z)) $
      dropWhile isDigit xs

readDictWords :: String → IO (Set.Set String)
readDictWords fname = do
  rawRead ← readFile fname
  let dictMap = Set.fromList $ filter (λz → length z < 7) (lines rawRead)
  return $ dictMap

rotate1 [] = []
rotate1 (x : xs) = concat [xs, [x]]

decrypt :: [Integer] → [Integer] → [Integer]
decrypt [] cip = []
decrypt (x : xs) cip = (x' : (decrypt xs cip'))
  where cip' = rotate1 cip
    x' = xor x (head cip)

decrypt' :: [Integer] → [Integer] → String
decrypt' xs cip = map (chr ∘ fromIntegral) (decrypt xs cip)

scoreDecryption :: String → Set.Set String → Int
scoreDecryption dcrypt dict =
  let nwords = words dcrypt
    dwords = filter (λz → z `Set.member` dict) nwords
  in length dwords

ciphers = [[a, b, c] | a ← [97..122], b ← [97..122], c ← [97..122]]

main = do
  dWords ← readDictWords "/usr/share/dict/words"
  rawEncrypted ← readEncrypted "cipher1.txt"
  let unencrypted = map (λz → (z, scoreDecryption (decrypt' rawEncrypted z) dWords)) ciphers
    bestKey = fst $ maximumBy (λx y → compare (snd x) (snd y)) unencrypted
    decrypted = decrypt' rawEncrypted bestKey
    sumAscii = sum $ map ord decrypted
  putStrLn $ "The sum of the ascii values of the decrypted message " ⧺
    "is " ⧺ show sumAscii ⧺ ".\nThe unencrypted message " ⧺
    "itself is \n\n" ⧺ decrypted
```

3 Result

```
ghc --make -O2 -optc-O3 problem59.lhs
./problem59
```

```
The sum of the ascii values of the decrypted message is 107359.
The unencrypted message itself is

(The Gospel of John, chapter 1) 1 In the beginning the Word already existed.
He was with God, and he was God. 2 He was in the beginning with God. 3 He
created everything there is. Nothing exists that he didn’t make. 4 Life itself
was in him, and this life gives light to everyone. 5 The light shines through
the darkness, and the darkness can never extinguish it. 6 God sent John the
Baptist 7 to tell everyone about the light so that everyone might believe
because of his testimony. 8 John himself was not the light; he was only a
witness to the light. 9 The one who is the true light, who gives light to
everyone, was going to come into the world. 10 But although the world was made
through him, the world didn’t recognize him when he came. 11 Even in his own
land and among his own people, he was not accepted. 12 But to all who believed
him and accepted him, he gave the right to become children of God. 13 They are
reborn! This is not a physical birth resulting from human passion or plan,
this rebirth comes from God.14 So the Word became human and lived here on
earth among us. He was full of unfailing love and faithfulness. And we have
seen his glory, the glory of the only Son of the Father.
```