1 Problem

The arithmetic sequence, 1487, 4817, 8147, in which each of the terms increases by 3330, is unusual in two ways:

1. each of the three terms are prime, and,

2. each of the 4-digit numbers are permutations of one another.

There are no arithmetic sequences made up of three 1-, 2-, or 3-digit primes, exhibiting this property, but there is one other 4-digit increasing sequence.

What 12-digit number do you form by concatenating the three terms in this sequence?

2 Solution

```
import Data.List
import qualified Data.Map as Map
import Data.Maybe
import System.Environment
import Data.Numbers
import Data.Numbers.Primes
import qualified Data.Set as Set

sPrimes = filter (>= 1000) $ takeWhile (<10000) primes

permPrimes x =
  let perms = map (\z -> read z :: Integer) (permutations $ show x)
  in sort $ filter isPrime perms

isArithSeq xs = and (map (\z -> z == (diffs !! 0)) diffs)
  where diffs = [(xs !! k) - (xs !! (k - 1)) | k <- [1 .. (length xs)]]

arithSeqs xs =
  let rots = rotations xs
      pairs = map (\z -> (head z, filter (\zzz -> zzz `elem` xs) $ map (\zz -> 0 - (head z) + 2 * zz) (filter (>= (head
      solns = filter (\z -> (snd z /= [])) pairs
      osols = (Set.toList . Set.fromList) $ map (\z -> sort ([fst z, head (snd z)])) solns
      fsols = map (\z -> (head z, (head z) + ((last z) - (head z) `div` 2), last z)) osols
  in fsols

rotations x = map (\z -> rotate z x) [0 .. (length x - 1)]
  where rotate n x = concat [drop n x, take n x]

main = do
  let ssprimes = sortBy (\x y -> compare (sort $ show x) (sort $ show y)) sPrimes
      gprimes = groupBy (\x y -> (sort $ show x) == (sort $ show y)) ssprimes
      lgprimes = filter (\z -> length z >= 3) gprimes
      triplets = concat $ map (\z -> [sort [x, y, 2 * y - x] | x <- z, y <- z, y > x]) lgprimes
      p1       = filter (\z -> ((z !! 2) `elem` sPrimes)) triplets
      p2       = filter (\z -> (sort $ show $ head z) == (sort $ show $ last z)) p1
      p3       = filter (\z -> 1487 `notElem` z) p2
      soln     = concat (map show (p3 !! 0))
  putStrLn $ soln
```

3 Result

```
runhaskell problem49.lhs
296962999629
```