1 Problem

An irrational decimal fraction is created by concatenating the positive integers:

$$0.12345678910\mathbf{1}12131415161718192021\ldots$$

It can be seen that the 12^{th} digit of the fractional part is 1.

If d represents the n^{th} digit of the fractional part, find the value of the following expression.

$$d_1 \times d_{10} \times d_{100} \times d_{1000} \times d_{10000} \times d_{100000} \times d_{1000000}$$

2 Solution

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

concatProduct :: Integral a => a -> [a] -> a
concatProduct x ys = fromIntegral (read pr :: Integer)
  where pr = (concat $ map (\z -> show (x * z)) ys)

class Digital a where
  isPandigital :: a -> Bool

instance Digital Integer where
  isPandigital a = (sort . show) a == (take (length $ show a) "123456789")

getDigit :: Integer -> Char
getDigit x =
  let ordr = length (takeWhile (<x) orderDigits)
      dff  = x - (orderDigits !! (fromIntegral ordr - 1))
      ndigs = length $ show $ 10 ^ ordr
      pos  = dff `mod` (fromIntegral $ ndigs)
      ns   = show (10 ^ ordr + (dff `div` fromIntegral ndigs))
  in (ns !! (fromIntegral pos))

orderDigits :: [Integer]
orderDigits = map (\z -> fromIntegral $ (+) 1 $ length $ concatMap show [1..(z-1)]) $ map (10^) [1..6]

main = do
  let powersTen = map (\z -> 10 ^ z) [2..6]
      digits    = map (\z -> read ((getDigit z) : "") :: Int) powersTen
      dProduct = product $ concat [[1,1], digits]
  putStrLn $ "The product of the specified digits is " ++ show dProduct ++ "."
```

3 Result

```
runhaskell problem40.lhs
The product of the specified digits is 210.
```