

1 Problem

145 is a curious number, as $1! + 4! + 5! = 1 + 24 + 120 = 145$.

Find the sum of all numbers which are equal to the sum of the factorial of their digits.

Note: as $1! = 1$ and $2! = 2$ are not sums they are not included.

2 Solution

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

factMap = Map.fromList
  [(0,1)
  ,(1,1)
  ,(2,2)
  ,(3,6)
  ,(4,24)
  ,(5,120)
  ,(6,720)
  ,(7,5040)
  ,(8,40320)
  ,(9,362880)
  ]

sumProduct :: Integer -> Integer
sumProduct x
  | x < 10    = fromJust $ Map.lookup x factMap
  | otherwise = fromJust (Map.lookup d factMap) + sumProduct x'
  where d = x `mod` 10
        x' = round $ fromInteger (x - x `mod` 10) / 10.0

intListToInt :: [Integer] -> Integer
intListToInt [] = 0
intListToInt (x : xs) = 10  $\uparrow$  (length xs) * x + intListToInt xs

intToIntList :: Integer -> [Integer]
intToIntList x
  | x < 10    = [x]
  | otherwise = concat [(intToIntList $ x - 10 * x `div` 10), [x]]

main = do
  let curInts = filter (\z -> z  $\equiv$  sumProduct z) [3..1000000]
  putStrLn $ "The only 'curious' integers of this type are " ++
    show curInts ++ " which sum to " ++ (show  $\circ$  sum) curInts ++ "."
```

3 Result

`runhaskell problem34.lhs`

The only 'curious' integers of this type are [145,40585] which sum to 40730.