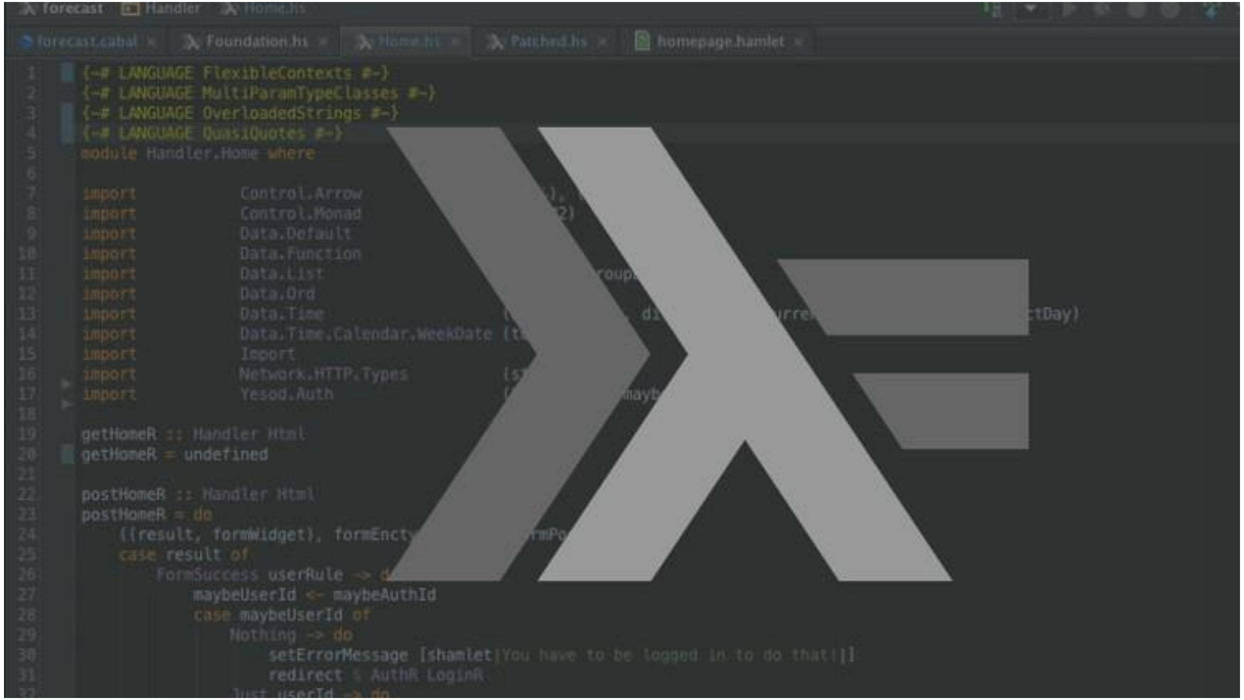
Infinite Lists Problems



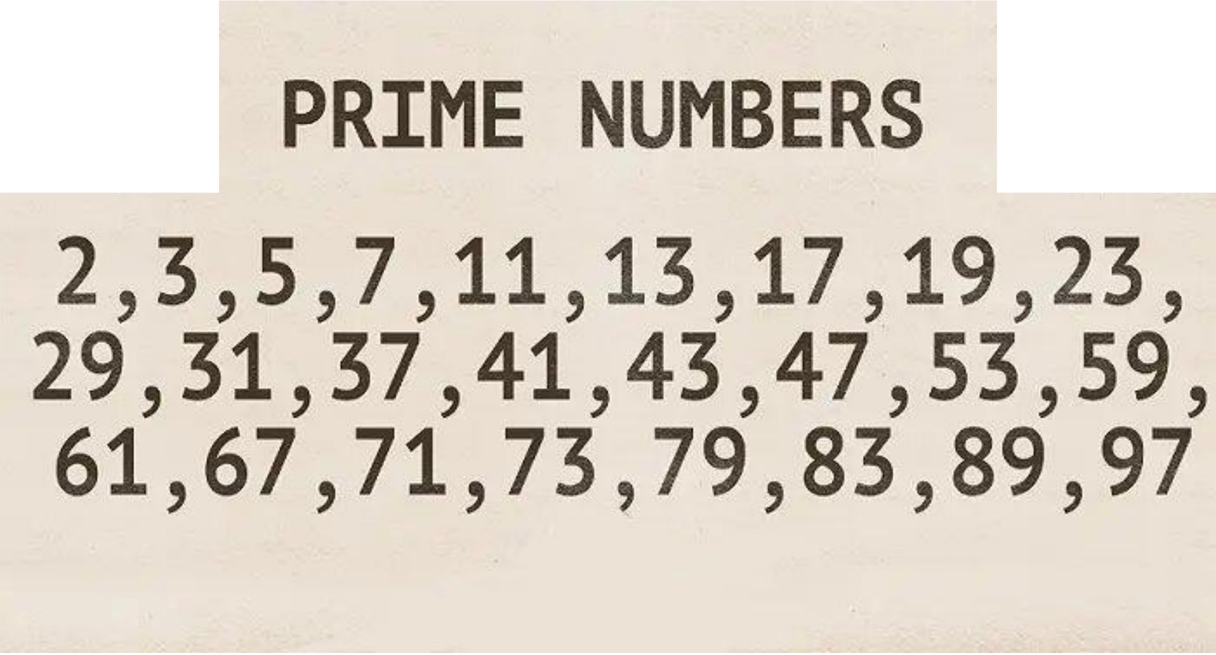
```haskell
{-# LANGUAGE FlexibleContexts #-}
{-# LANGUAGE MultiParamTypeClasses #-}
{-# LANGUAGE OverloadedStrings #-}
{-# LANGUAGE QuasiQuotes #-}
module Handler.Home where

import           Control.Arrow
import           Control.Monad
import           Data.Default
import           Data.Function
import           Data.List
import           Data.Ord
import           Data.Time
import           Data.Time.Calendar.WeekDate (t
import           Import
import           Network.HTTP.Types          (s
import           Yesod.Auth

getHomeR :: Handler Html
getHomeR = undefined

postHomeR :: Handler Html
postHomeR = do
    ((result, formWidget), formEncty
    case result of
        FormSuccess userRule -> d
            maybeUserId <- maybeAuthId
            case maybeUserId of
                Nothing -> do
                    setErrorMessage [shamlet|You have to be logged in to do that!|]
                    redirect $ AuthR LoginR
                Just userId -> do
```

Lucas Bazilio - Udemy

Problem 7

The goal of this problem is to work the definition of infinite lists. In particular, you are required to define the function that generates the sequence of the prime numbers [2,3,5,7,11,13,17,19…]. Use the function *primes :: [Integer]*

Input	Output
`take 6 primes`	`-> [2,3,5,7,11,13]`
`take 4 primes`	`-> [2,3,5,7]`



PRIME NUMBERS

2, 3, 5, 7, 11, 13, 17, 19, 23,
29, 31, 37, 41, 43, 47, 53, 59,
61, 67, 71, 73, 79, 83, 89, 97

A prime number is a whole number greater than 1 whose only factors are 1 and itself.

Instructor Youtube Channel: Lucas Science