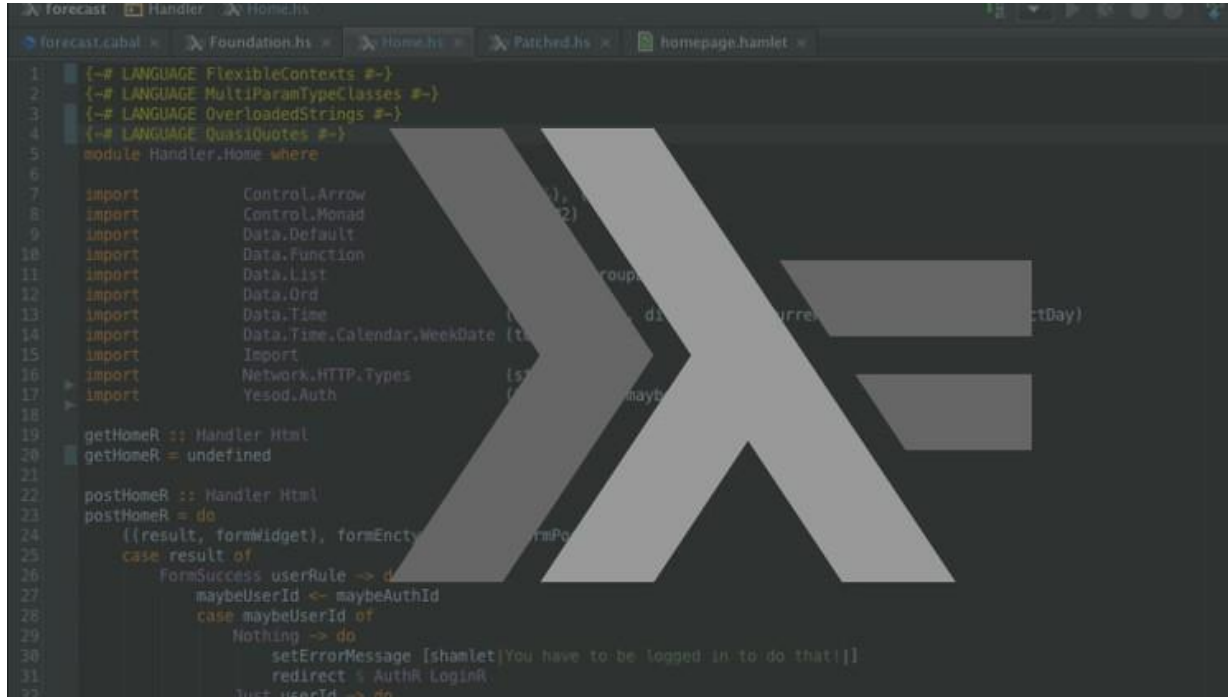


Infinite Lists Problems



```
1 {-# LANGUAGE FlexibleContexts #-}
2 {-# LANGUAGE MultiParamTypeClasses #-}
3 {-# LANGUAGE OverloadedStrings #-}
4 {-# LANGUAGE QuasiQuotes #-}
5 module Handler.Home where
6
7 import Control.Arrow
8 import Control.Monad
9 import Data.Default
10 import Data.Function
11 import Data.List
12 import Data.Ord
13 import Data.Time
14 import Data.Time.Calendar.WeekDate
15 import Import
16 import Network.HTTP.Types
17 import Yesod.Auth
18
19 getHomeR :: Handler Html
20 getHomeR = undefined
21
22 postHomeR :: Handler Html
23 postHomeR = do
24   ((result, formWidget), formEnctype) <- runFormPost
25   case result of
26     FormSuccess userRule -> do
27       maybeUserId <- maybeAuthId
28       case maybeUserId of
29         Nothing -> do
30           setErrorMessage [shamlet|You have to be logged in to do that!|]
31           redirect % AuthR.LoginR
32         Just userId -> do
```

Problem 8

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 ordered sequence of the Hamming numbers [1,2,3,4,5,6,8,9...]. The Hamming numbers are those that only have 2, 3 and 5 as prime divisors. Use the function *hammings* :: [Integer]

Input

Output

take 8 hammings

-> [1,2,3,4,5,6,8,9]

take 5 hammings

-> [1,2,3,4,5]

Instructor Youtube Channel: Lucas Science

