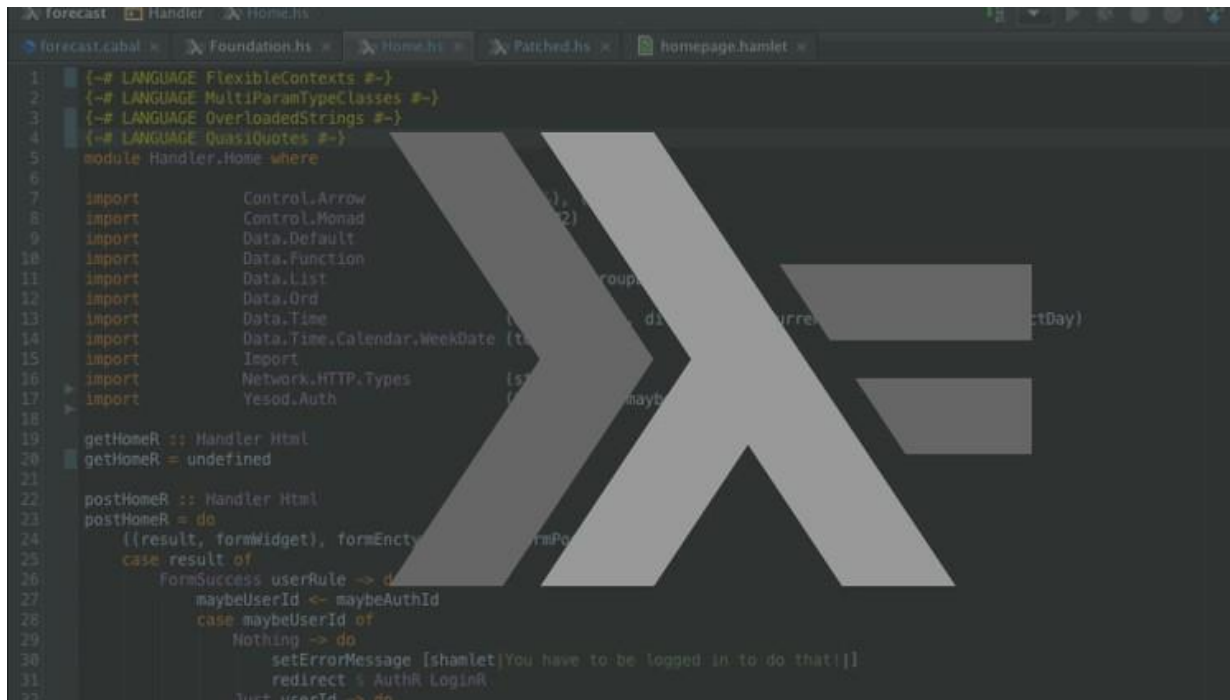


Higher Order Functions Problems



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

Problem 1

Implement a function $eq1 :: [Int] \rightarrow [Int] \rightarrow Bool$ that tells whether two lists of integers are equal.

Input

```
eq1 [1,2,3] [1,2,3]  
eq1 [1,2,3] [3,2,1]  
eq1 [1,2,3] [1,2,3,4]
```

Output

```
True  
False  
False
```

Instructor Youtube Channel: Lucas Science

