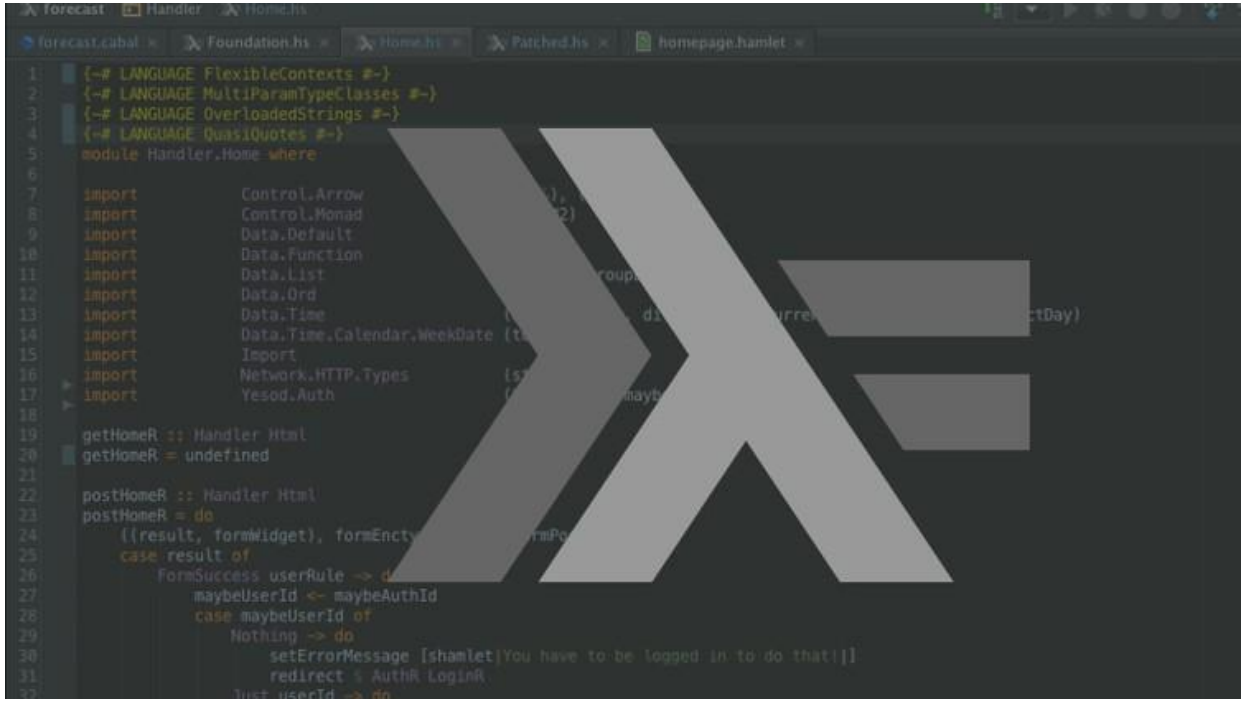
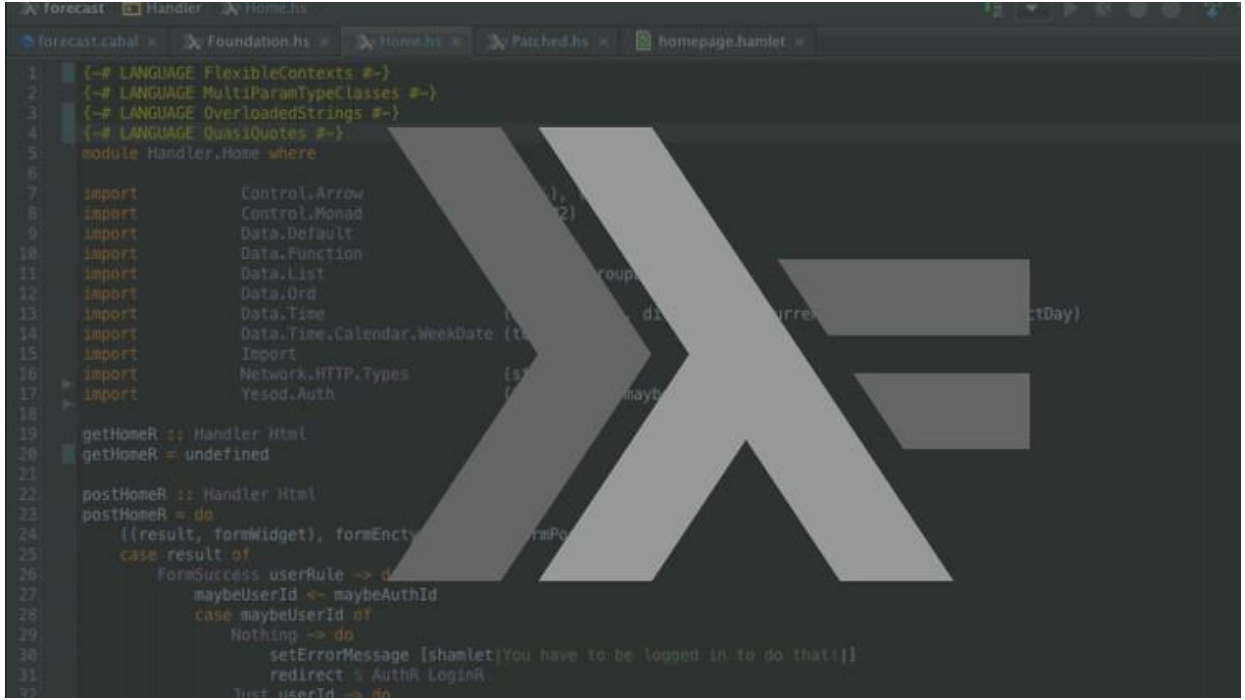
Input and Output Problems



Problem 5


forecast.cabal
Foundation.hs
Home.hs
Patched.hs
homepage.hamlet
{-# 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
import Import
import Network.HTTP.Types
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

Problem 5

In this problem, you are going to implement the "number guessing game" in Haskell. Here is an example of how this might work:

```
Think of a number between 1 and 100!
Is it 50? higher
Is it 75? lower
Is it 62? lower
Is it 56? yes
Great, I won!
```

Problem 5

Implement a function

game :: IO ()
That plays one instance of this game.

You might need the following predefined functions:

getLine :: IO String -- reads a line of user input
putStrLn :: String -> IO () -- outputs one line of text

Importance of Flushing

Flushing avoids this:

```
1
Enter some value: Enter another value:2
3
4
Enter a point this time: Enter x: Enter y: y
Is this correct? (y/n):
```