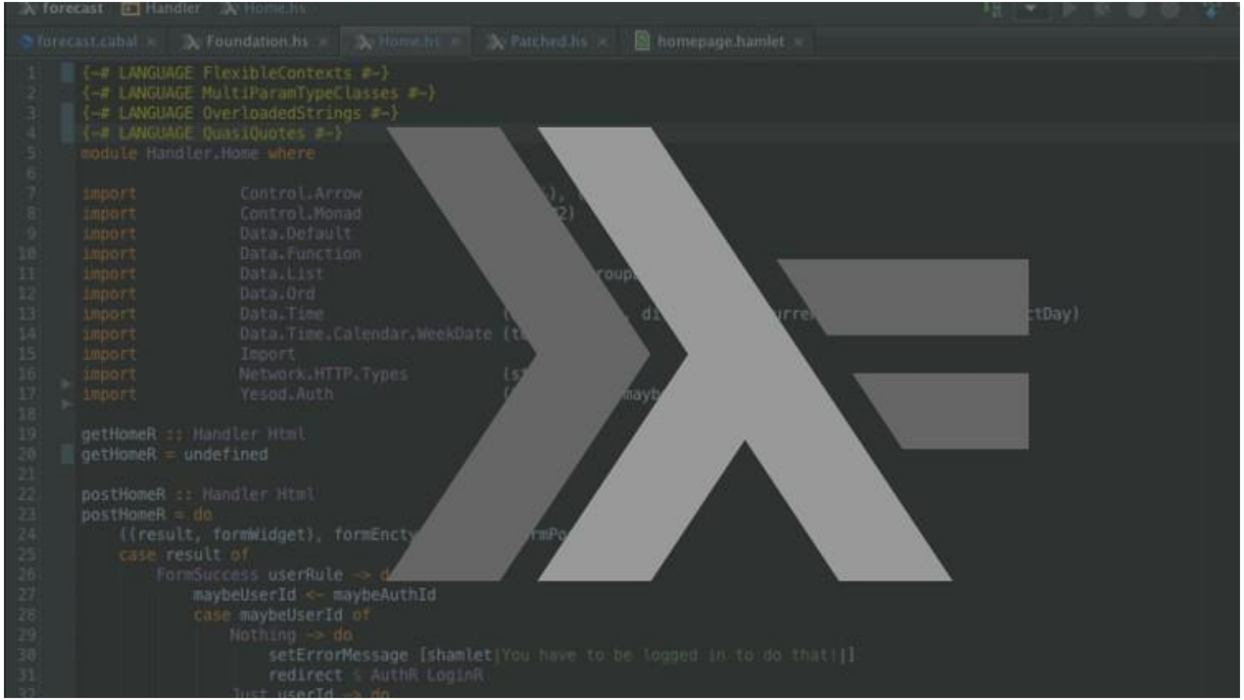
Infinite Lists Problems



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

import           Control.Arrow                    ), (
import           Control.Monad                    2)
import           Data.Default
import           Data.Function
import           Data.List                        roup
import           Data.Ord
import           Data.Time                 (         , di       urre                    tDay)
import           Data.Time.Calendar.WeekDate (t
import           Import
import           Network.HTTP.Types        (s
import           Yesod.Auth                (              mayb

getHomeR :: Handler Html
getHomeR = undefined

postHomeR :: Handler Html
postHomeR = do
    ((result, formWidget), formEncty            rmPo
    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

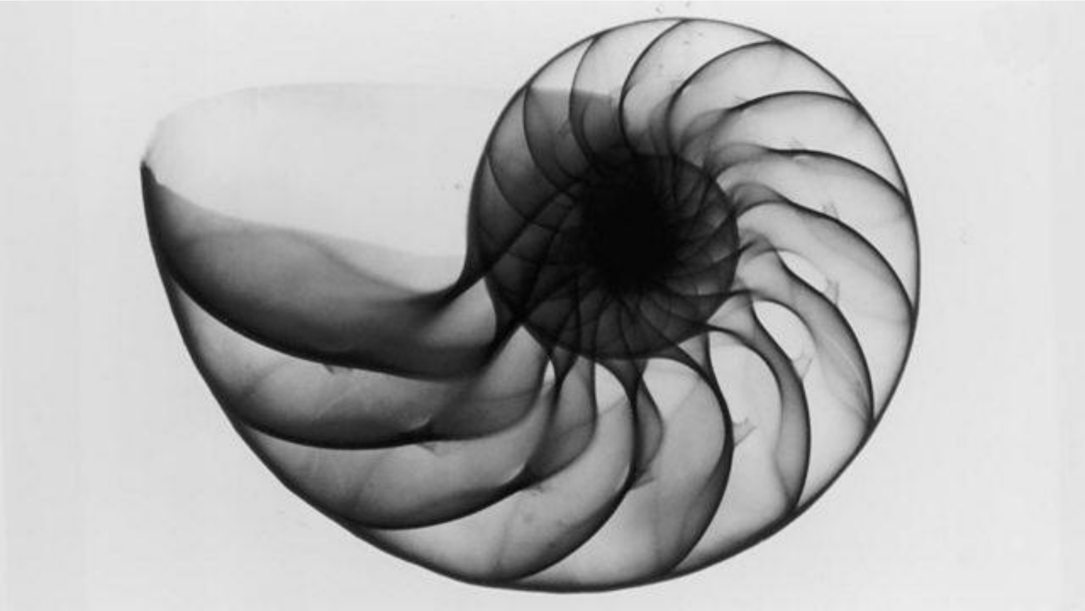
Problem 6

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 Fibonacci numbers [0,1,1,2,3,5,8,13…]. Use the function *fibs :: [Integer]*

Input	Output
`take 6 fibs`	`->  [0,1,1,2,3,5]`
`take 4 fibs`	`->  [0,1,1,2]`

Lucas Bazilio - Udemy

Fibonacci Sequence



each number is the sum of the two preceding ones

Lucas Bazilio - Udemy

Instructor Youtube Channel: Lucas Science