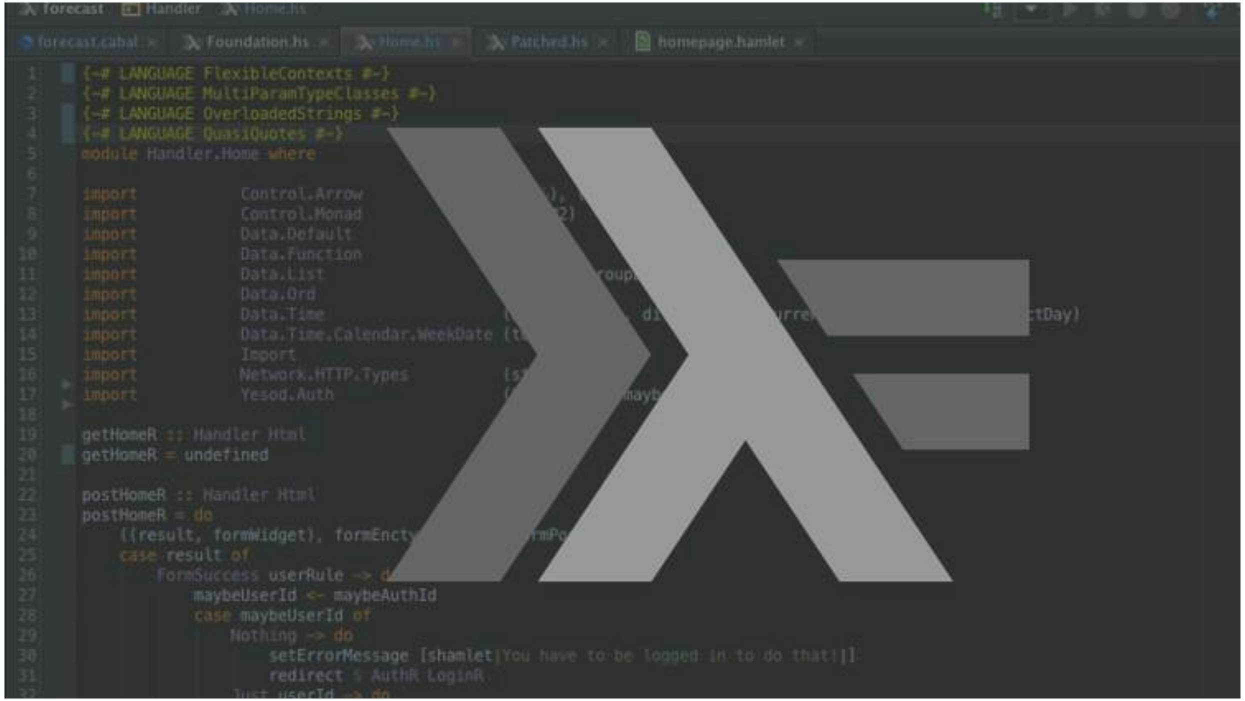
Higher Order Functions Problems



```
{-# 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 4

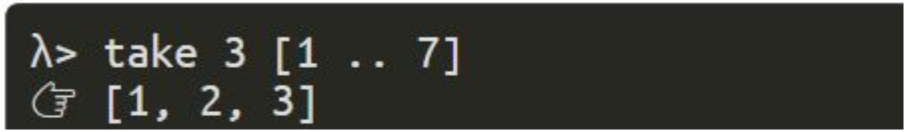
Implement a function *powersOf2 :: [Int]* that generates the list of all the powers of 2.

Input   Output

```
take 5 powersOf2                     ->  [1,2,4,8,16]

take 3 powersOf2                     ->  [1,2,4]
```

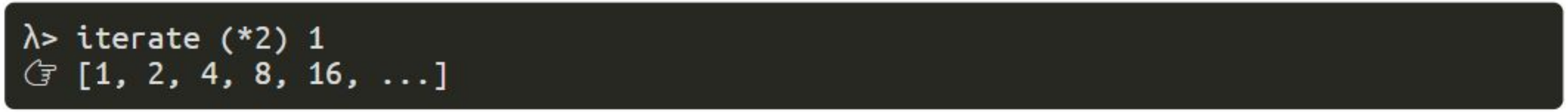
take



```
λ> take 3 [1 .. 7]
☞ [1, 2, 3]
```

Lucas Bazilio - Udemy

iterate



```
λ> iterate (*2) 1
☞ [1, 2, 4, 8, 16, ...]
```

Instructor Youtube Channel: Lucas Science

Lucas Science

SUBSCRIBE