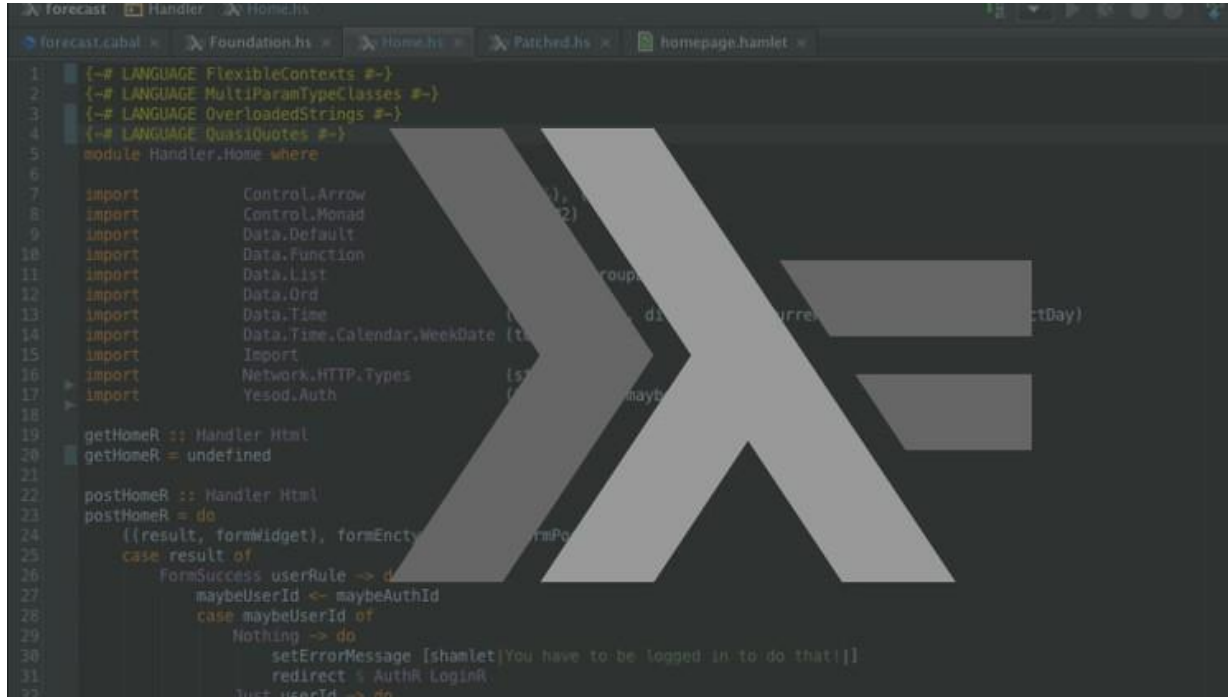


Infinite Lists Problems



```
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 4

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 triangular numbers $[0,1,3,6,10,15,21,28\dots]$. Use the function *triangulars* :: *[Integer]*

Input

Output

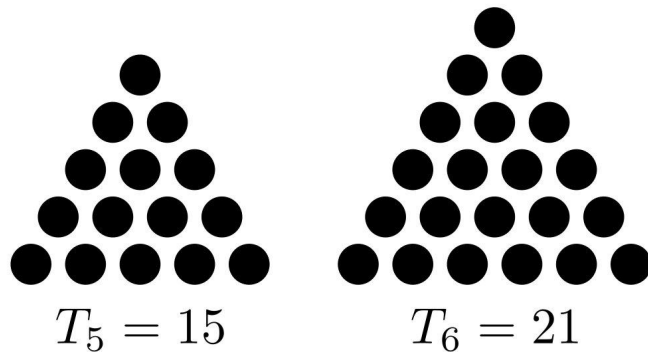
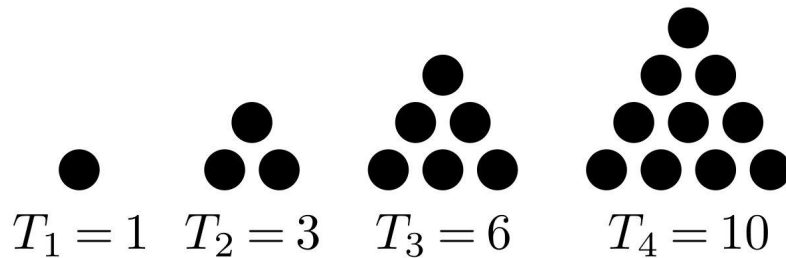
take 6 triangulars

-> $[0,1,3,6,10,15]$

take 4 triangulars

-> $[0,1,3,6]$

Triangular Numbers



Triangular Numbers

$$T_n = \sum_{k=1}^n k = 1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$$

iterate

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

scanl

Input: scanl (/) 64 [4,2,4]

Output: [64.0,16.0,8.0,2.0]

Instructor Youtube Channel: Lucas Science

