```
-- Homework 2

module Log where

import Control.Applicative

data MessageType = Info
                 | Warning
                 | Error Int
  deriving (Show, Eq)

type TimeStamp = Int

data LogMessage = LogMessage MessageType TimeStamp String
                | Unknown String
  deriving (Show, Eq)

data MessageTree = Leaf
                 | Node MessageTree LogMessage MessageTree
    deriving (Show, Eq)

-- | @testParse p n f@ tests the log file parser @p@ by running it
--   on the first @n@ lines of file @f@.
testParse :: (String -> [LogMessage])
          -> Int
          -> FilePath
          -> IO [LogMessage]
testParse parse n file = take n . parse <$> readFile file

-- | @testWhatWentWrong p w f@ tests the log file parser @p@ and
--   warning message extractor @w@ by running them on the log file
--   @f@.
```

```
testWhatWentWrong :: (String -> [LogMessage])
                  -> ([LogMessage] -> [String])
                  -> FilePath
                  -> IO [String]
testWhatWentWrong parse whatWentWrong file
  = whatWentWrong . parse <$> readFile file
```