

```

-- Model solution for Lab07, Ex 2
--
-- Author: Manuel M T Chakravarty

module Main
where

import Prelude hiding (lines)
import IO

-- Split a string containing '\n' characters into a list of strings - one per
-- line without the newline characters
--
lines      :: String -> [String]
lines str  = let
    (line, rest) = oneLine str
  in
    if rest == ""
    then
      if line == "" then [] else [line]
    else
      line : lines rest
  where
    --
    -- extract a single line (terminated by either '\n' or the end
    -- of the string)
    --
    oneLine      :: String -> (String, String)
    oneLine ""   = ("", "")
    oneLine ('\n':cs) = ("", cs)
    oneLine (c  :cs) = let
        (line, rest) = oneLine cs
      in
        (c:line, rest)

-- add line numbers to a list of strings
--
addLineNumber      :: Int -> [String] -> [String]
addLineNumber n [] = []
addLineNumber n (line:lines) =
  (show n ++ ": " ++ line) : addLineNumber (n + 1) lines

main :: IO ()
main = do
  hSetBuffering stdout NoBuffering
  putStr "Enter filename: "
  fname <- getLine
  contents <- readFile fname
  putStr (unlines (addLineNumber 1 (lines contents)))

```