```haskell
-- Model solution for Tut04
--
-- Copyright [2000..2004] Manuel M T Chakravarty

module Tut04
where

import Prelude hiding (sum, enumFromTo)


-- Compute the factorial of a positive number
--
-- Example: fact 6 = 720
--
fact                :: Int -> Int
fact 1               = 1
fact n  | n > 1      = n * fact (n - 1)
        | otherwise  = error "fact: input too small"

-- Enumerate a list of numbers in a given range
--
-- Examples: enumFromTo 3 10 = [3,4,5,6,7,8,9,10]
--           enumFromTo 4 2  = []
--
enumFromTo                  :: Int -> Int -> [Int]
enumFromTo m n | m > n       = []
               | otherwise   = m : enumFromTo (m + 1) n

-- sum the numerals within a list
--
-- Example: sum [5, 3, 1] = 9
--
sum         :: Num a => [a] -> a
sum []       = 0
sum (x:xs)   = x + sum xs

-- remove all odd integers from a list
--
-- Example: removeOdd [1, 2, 3, 4, 5] = [2, 4]
--
removeOdd                    :: [Int] -> [Int]
removeOdd []                  = []
removeOdd (x:xs) | odd x      = removeOdd xs
                 | otherwise  = x : removeOdd xs
```