

```

-- Model solution for Tut03
--
-- Copyright [2000..2004] Manuel M T Chakravarty

module Tut03
where

-- Given two integer numbers, return a pair with the smaller number first
--
-- Example: sort2 10 20 = (10, 20)
--          sort2 15 5  = (5, 15)
--
sort2 :: Int -> Int -> (Int, Int)
sort2 x y | x < y      = (x, y)
          | otherwise  = (y, x)

-- Compute interest on an account, where the interest rates are as follows:
--
-- * 0% per year on the first $999
-- * 2% on every dollar from $1000 to $4,999
-- * 4% for every dollar from $5000 to $19,999
-- * 5% for every dollar from $20,000
--
-- Example: interest 4000 = 60
--          interest 40000 = 1680.0
--
interest :: Float -> Float
interest amount
  | amount < 1000    = 0
  | amount < 5000   = (amount - 1000) * 0.02
  | amount < 20000  = (5000 - 1000) * 0.02 + (amount - 5000) * 0.04
  | otherwise       = (5000 - 1000) * 0.02 + (20000 - 5000) * 0.04
                    + (amount - 20000) * 0.05

-- As before but using local definitions
--
interest' :: Float -> Float
interest' amount
  | amount < 1000    = 0
  | amount < 5000   = (amount - 1000) * 0.02
  | amount < 20000  = (amount - 5000) * 0.04 + step1
  | otherwise       = (amount - 20000) * 0.05 + step2
  where
    step1 = (5000 - 1000) * 0.02
    step2 = (20000 - 5000) * 0.04 + step1

-- Definition of a colour point from the lecture
--
type Colour = String

type ColourPoint = (Int, Int, Colour)

-- Given a colour point and colour, change the colour of the point
-- correspondingly
--
-- Example: changeColour (10, 20, "white") "black" = (10, 20, "black")
--
changeColour :: ColourPoint -> Colour -> ColourPoint
changeColour (x, y, oldColour) newColour = (x, y, newColour)

-- Test whether the position of two points is the same
--
-- Example: equalPos (10, 20, "yellow") (10, 20, "green") = True
--          equalPos (11, 20, "yellow") (10, 20, "yellow") = False
--
equalPos :: ColourPoint -> ColourPoint -> Bool
equalPos (x1, y1, _) (x2, y2, _) = x1 == x2 && y1 == y2

```

-- Answer to Q4: Because we know that Cartesian coordinates in a
-- two-dimensional plane **always** require two components.
-- Whenever, the number of components is fixed like this, a
-- tuple is usually preferred over a list.