```haskell
-- Model solution for Lab03
--
-- Copyright [2000..2004] Manuel M T Chakravarty

module Lab03
where

-- Determine whether a number of positive
--
-- Example: isPositive 10 = True
--          isPositive -2 = False
--
isPositive               :: Int -> Bool
isPositive x | x > 0      = True
             | otherwise  = False

-- A shorter and more concise version (this one is better)
--
isPositive'   :: Int -> Bool
isPositive' x  = x > 0

-- Given three integers, produce a triple whether the integers occur in
-- increasing order
--
-- Example: sort3 1 2 3 = (1, 2, 3)
--          sort3 2 1 3 = (1, 2, 3)
--          sort3 2 3 1 = (1, 2, 3)
--          sort3 3 2 1 = (1, 2, 3)
--          sort3 3 1 2 = (1, 2, 3)
--
sort3                                 :: Int -> Int -> Int -> (Int, Int, Int)
sort3 x y z  | xSmallerY && ySmallerZ  = (x, y, z)
             | xSmallerY && xSmallerZ  = (x, z, y)
             | xSmallerZ && ySmallerZ  = (y, x, z)
             | xSmallerY               = (z, x, y)
             | ySmallerZ               = (y, z, x)
             | otherwise               = (z, y, x)
                                         where
                                           xSmallerY = x < y
                                           ySmallerZ = y < z
                                           xSmallerZ = x < z

-- Function from the lecture
--
sort2                 :: Int -> Int -> (Int, Int)
sort2 x y | x < y      = (x, y)
          | otherwise  = (y, x)

-- Variant of sort3
--
sort3'       :: Int -> Int -> Int -> (Int, Int, Int)
sort3' x y z  = (smallest, middle, greatest)
  where
    (smallOrMiddle, middleOrGreat1) = sort2 x y
    (smallest     , middleOrGreat2) = sort2 smallOrMiddle z
    (middle       , greatest      ) = sort2 middleOrGreat1 middleOrGreat2

-- Interval type
--
type Interval = (Int, Int)

-- Compute a range of integers with a fixed increment
--
-- Example: rangeWithInc (10, 21) 2    = [10,12,14,16,18,20]
--          rangeWithInc (15, 1) (-1) = [15,14,13,12,11,10,9,8,7,6,5,4,3,2,1]
--
rangeWithInc                      :: Interval -> Int -> [Int]
rangeWithInc (from, to) increment  = [from, from + increment..to]
```

```
-- Determine whether both arguments are ‘True’
--
-- Example: bothTrue False False = False
--          bothTrue True  False = False
--          bothTrue False True  = False
--          bothTrue True  True  = True
--
bothTrue                 :: Bool -> Bool -> Bool
bothTrue a b | a          = b
             | otherwise  = False
```