[Browse source program] [Execute source program]

```
-- Simple examples of list comprehensions

squares   = [x * x | x <- [0 ..]]
triangles = [x * (x+1) `div` 2 | x <- [0 ..]]

-- take 20 squares   => [0,1,4,9,16,25,36,49,64,81]
-- take 10 triangles => [0,1,3,6,10,15,21,28,36,45]

primes = [x | x <- [2 ..], isPrime x]
  where isPrime x | x >= 2 = not (hasFactorsFrom 2)
          where hasFactorsFrom y | y*y > x    = False
                                 | otherwise  = x `mod` y == 0 ||
                                                hasFactorsFrom (y+1)

-- take 10 primes => [2,3,5,7,11,13,17,19,23,29]

lexPairs = [(x,y) | x <- [0 .. 3], y <- [x .. 3]]

-- lexPairs => [(0,0),(0,1),(0,2),(0,3),(1,1),(1,2),(1,3),(2,2),(2,3),(3,3)]
```