Practice with Arrays

1) def lengths xs
   xs.map { |x| x.length }
end

2) def rev xs
   xs.inject([]) { |acc, x| acc.unshift x }
end

3) def num_even xs
   xs.count { |x| x.even? }
end

4) def all_equal? xs
   xs.all? { |x| x == xs[0] }
end

5) def prime n
   (2..n).select { |x| (2..Math.sqrt(x).to_i).all? { |y| x % y != 0 } }
end

6) def trigger_sum xs, x
   sub = false
   xs.inject(0) do |acc, y|
     if sub or x == y
       sub = true acc - y
     else acc + y
     end
   end
end
Practice with Hashes

1) def keys_and_values h
   h.keys & h.values
   end

2) def flip_hash h
   res = {}
   h.each {|k, v| res[v] = k } res
   end

3) def intersect h1, h2
   h1.select {|k, v| h2[k] == v }
   end

Practice using blocks

1) def our_map1 xs
   result = Array.new(xs.length)
   i = 0
   xs.each do |x|
     result[i] = yield x
     i += 1
   end
   result
   end

def our_map2 xs
   result = []
   xs.each {|x| result.push yield x }
   result
end

2) def our_select xs
   result = []
   xs.each {|x| result.push x if yield x }
   result
   end

3) def our_inject xs, init
   xs.each {|x| init = yield init, x }
   init
   end