def new\_array(array)

p "new\_array", array.object\_id

array.map{|ele| ele.upcase}.object\_id

end

def mutate\_array!(array)

p "mutate\_array", array.object\_id

array.map!{|ele| ele.upcase}.object\_id

end

# p new\_array(("a".."z").to\_a)

# p mutate\_array!(("a".."z").to\_a)

#array grid

def grid(n, m)

# Array.new(n, Array.new(m)) #creates subarrays with the same object\_id

Array.new(n) { Array.new(m,0) } #creates unique subarrays

end

# p grid(4, 5)

#hash count

def element\_count(array)

count = Hash.new(0)

# count = Hash.new {|hash, key| hash[key] = 0}

array.each { |el| count[el] += 1 }

count

end

# p element\_count(["a", "b","b", "a"])

def hash\_indices(array)

# hash = Hash.new([]) #creates a default of array with same object\_id

hash = Hash.new {|hash, key| hash[key] = []} #creates default of unique array

array.each\_with\_index { |el, i| hash[el] << i }

hash

end

# p hash\_indices(["a", "b","b", "a"])