# A4\_Stryshak.py - a class for implementing warm-up functions

# @author Erik Stryshak

# @version 1.0

from random import randint

# takes in a list and the two indices to swap

def swap(list, index\_one, index\_two):

# stores the value of index\_one in a variable and swaps the values

first\_value = list[index\_one]

list[index\_one] = list[index\_two]

list[index\_two] = first\_value

return list

# takes in a list and uses the swap function to swap each index with

# a random index

def shuffle(list):

for index, item in enumerate(list):

swap(list, index, randint(0,len(list)-1))

return list

# takes in a list and returns False if the number at an index is less

# than the number at the previous index

def isSortedArray(list):

for index, item in enumerate(list):

if(index == 0):

index = 0

elif(list[index] < list[index-1]):

return False

else:

index = index

return True

# takes in a list and compares the length of the list to the length of

# a set that is passed the same list. the set filters out unique values

def isUniqueValuesArray(list):

if(len(list) > len(set(list))):

return False

return True

# uses the isSortedArray and isUniqueValuesArray functions

def isSortedAndUniqueArray(list):

if(isSortedArray(list)):

if(isUniqueValuesArray(list)):

return True

return False

# takes in the size of the list to be generated

def genSortedArrayUniqueValues(size):

list\_to\_return = []

# add a random number between 1 and 5 to the list

list\_to\_return.append(randint(1,5))

i = 1

# keeping incrementing by a random integer between 1 and 5 until

# the array is of the requested size

while(i<size):

num\_prev = list\_to\_return[i-1]

num\_inc = randint(1,5)

list\_to\_return.append(num\_prev + num\_inc)

i = i + 1

return list\_to\_return

def areEqualLists(list\_one, list\_two):

if(len(list\_one) != len(list\_two)):

return False

for index in range(len(list\_one)):

if(list\_one[index] != list\_two[index]):

return False

return True

def main():

#SWAP TEST

test = [1,2,3,4,5,6]

swap\_test\_expected = [6,2,3,4,5,1]

test = swap(test,0,5)

if(areEqualLists(test, swap\_test\_expected)):

print("Swap test PASSED")

else:

print("Swap test FAILED")

#SHUFFLE TEST

test = [1,2,3,4,5,6]

test\_original = [1,2,3,4,5,6]

test = shuffle(test)

if(areEqualLists(test\_original, test)):

print("Shuffle test FAILED. List was originally: ")

for p in test\_original: print(str(p) + ",", end = '')

print(" and after shuffling is: ", end = '')

for p in test: print(str(p) + ",", end = '')

print()

else:

print("Shuffle test PASSED. List was originally: ", end='')

for p in test\_original: print(str(p) + ",", end = '')

print(" and after shuffling is: ",end = '')

for p in test: print(str(p) + ",", end = '')

print()

#IS SORTED TEST

sorted\_list\_one = [1,2,3,4]

sorted\_list\_two = [1,1,2,2,2,4]

unsorted\_list = [5,1,3,7,2]

expected = True

#first sorted array

if(expected == isSortedArray(sorted\_list\_one)):

print("isSortedArray PASSED with list: ", end = '')

for p in sorted\_list\_one: print(str(p) + ",", end = '')

else:

print("isSortedArray FAILED with list: ", end = '')

for p in sorted\_list\_one: print(str(p) + ",", end = '')

#second sorted array

print()

if(expected == isSortedArray(sorted\_list\_two)):

print("isSortedArray PASSED with list: ", end = '')

for p in sorted\_list\_two: print(str(p) + ",", end = '')

else:

print("isSortedArray FAILED with list: ", end = '')

for p in sorted\_list\_two: print(str(p) + ",", end = '')

#unsorted array test

print()

if(expected == isSortedArray(unsorted\_list)):

print("isSortedArray PASSED with list: ", end = '')

for p in unsorted\_list: print(str(p) + ",", end = '')

else:

print("isSortedArray FAILED with list: ", end = '')

for p in unsorted\_list: print(str(p) + ",", end = '')

print()

#IS UNIQUE TEST

unique\_list = [1,2,3,4,5]

not\_unique\_list = [2,2,3,4]

expected = True

#unique test

if(expected == isUniqueValuesArray(unique\_list)):

print("isUniqueValuesArray PASSED with: ", end='')

for p in unique\_list: print(str(p) + ",", end='')

else:

print("isUniqueValuesArray FAILED with: ", end='')

for p in unique\_list: print(str(p) + ",", end='')

print()

#not unique test

if(expected == isUniqueValuesArray(not\_unique\_list)):

print("isUniqueValuesArray PASSED with: ", end='')

for p in not\_unique\_list: print(str(p) + ",", end='')

else:

print("isUniqueValuesArray FAILED with: ", end='')

for p in not\_unique\_list: print(str(p) + ",", end='')

print()

#IS SORTED AND UNIQUE TEST

sorted\_unique = [1,2,3,4]

sorted\_not\_unique = [1,2,2,3]

unique\_not\_sorted = [8,5,9]

not\_unique\_not\_sorted = [8,8,5,9]

#sorted unqique

if(isSortedAndUniqueArray(sorted\_unique)):

print("isSortedAndUniqueArray PASSED with: ", end='')

for p in sorted\_unique: print(str(p) + ",", end='')

else:

print("isSortedAndUniqueArray FAILED with: ", end='')

for p in sorted\_unique: print(str(p) + ",", end='')

# FAILURE 1

print()

if(isSortedAndUniqueArray(sorted\_not\_unique)):

print("isSortedAndUniqueArray PASSED with: ", end='')

for p in sorted\_not\_unique: print(str(p) + ",", end='')

else:

print("isSortedAndUniqueArray FAILED with: ", end='')

for p in sorted\_not\_unique: print(str(p) + ",", end='')

#FAILURE 2

print()

if(isSortedAndUniqueArray(unique\_not\_sorted)):

print("isSortedAndUniqueArray PASSED with: ", end='')

for p in unique\_not\_sorted: print(str(p) + ",", end='')

else:

print("isSortedAndUniqueArray FAILED with: ", end='')

for p in unique\_not\_sorted: print(str(p) + ",", end='')

#FAILURE 3

print()

if(isSortedAndUniqueArray(not\_unique\_not\_sorted)):

print("isSortedAndUniqueArray PASSED with: ", end='')

for p in not\_unique\_not\_sorted: print(str(p) + ",", end='')

else:

print("isSortedAndUniqueArray FAILED with: ", end='')

for p in not\_unique\_not\_sorted: print(str(p) + ",", end='')

print()

# GEN SORTED ARRAY UNIQUE VALUES TEST

gen\_list\_one = genSortedArrayUniqueValues(5)

gen\_list\_two = genSortedArrayUniqueValues(10)

if(isSortedAndUniqueArray(gen\_list\_one)):

print("genSortedArrayUniqueValues PASSED with: ", end='')

for p in gen\_list\_one: print(str(p) + ",", end='')

else:

print("genSortedArrayUniqueValues FAILED with: ", end='')

for p in gen\_list\_one: print(str(p) + ",", end='')

print()

if(isSortedAndUniqueArray(gen\_list\_two)):

print("genSortedArrayUniqueValues PASSED with: ", end='')

for p in gen\_list\_two: print(str(p) + ",", end='')

else:

print("genSortedArrayUniqueValues FAILED with: ", end='')

for p in gen\_list\_two: print(str(p) + ",", end='')

if \_\_name\_\_ == "\_\_main\_\_":

main()

OUTPUT:

