```
# 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:

```
C:\Users\Rik\Desktop\Programming_Languages\Python_Warmup>python A4_Stryshak.py
Swap test PASSED
Shuffle test PASSED. List was originally: 1,2,3,4,5,6, and after shuffling is: 3,6,5,4,1,2,
isSortedArray PASSED with list: 1,2,3,4,
isSortedArray PASSED with list: 1,1,2,2,2,4,
isSortedArray FAILED with list: 5,1,3,7,2,
isUniqueValuesArray PASSED with: 1,2,3,4,5,
isUniqueValuesArray FAILED with: 2,2,3,4,
isSortedAndUniqueArray PASSED with: 1,2,3,4,
isSortedAndUniqueArray FAILED with: 1,2,2,3,
isSortedAndUniqueArray FAILED with: 8,5,9,
isSortedAndUniqueArray FAILED with: 8,8,5,9,
genSortedArrayUniqueValues PASSED with: 1,6,7,12,16,
genSortedArrayUniqueValues PASSED with: 4,6,8,11,16,17,18,23,25,30,
```