**Problem 1:**

**import** random  
  
  
**def gen\_sequence**(length):  
 seq = ""  
  
 **for** i **in** range(length):  
 seq += str(random.randint(0, 9))  
  
 **return** seq  
  
**def gen\_sequence\_list**(length, count):  
 list = []  
  
 **for** i **in** range(count):  
 list.append(gen\_sequence(length))  
  
 **return** list  
  
  
random.seed()  
  
seq\_length = int(raw\_input("How long would you like your sequences to be? "))  
seq\_count = int(raw\_input("How many sequences would you like? "))  
**print**list = gen\_sequence\_list(seq\_length, seq\_count)  
  
**for** i **in** range(len(list)):  
 **print** list[i]

How long would you like your sequences to be? 6

How many sequences would you like? 7

431076

317642

018612

832917

208463

342585

575757

**Problem 2:**

**def seqzip**(seq1, seq2):  
 master\_list = []  
  
 min\_length = len(min(seq1, seq2))  
  
 **for** i **in** range(min\_length):  
 master\_list.append([seq1[i], seq2[i]])  
  
 **return** master\_list  
  
  
list = [1, 2, 3, 4, "Hello World", ["one", 1], ("tuple", 3)]  
tup = ('a', 'b', 'c', 'd', "Hey World", ["two", 2], ("Tuple", 4), "N/A")  
  
**print** "List: " + str(list)  
**print** "Tuple: " + str(tup)  
  
**print** "seqzip(List, Tuple) = " + str(seqzip(list, tup))

List: [1, 2, 3, 4, 'Hello World', ['one', 1], ('tuple', 3)]

Tuple: ('a', 'b', 'c', 'd', 'Hey World', ['two', 2], ('Tuple', 4), 'N/A')

seqzip(List, Tuple) = [[1, 'a'], [2, 'b'], [3, 'c'], [4, 'd'],

['Hello World', 'Hey World'], [['one', 1], ['two', 2]],

[('tuple', 3), ('Tuple', 4)]]

**Problem 3:**

Problem 1:

**import** random  
  
  
**def gen\_sequence**(length):  
 seq = ""  
  
 **for** i **in** range(length):  
 seq += str(random.randint(0, 9))  
  
 **return** seq  
  
  
**def gen\_sequence\_list**(length, count):  
 list = []  
  
 **for** i **in** range(count):  
 list.append(gen\_sequence(length))  
  
 **return** list  
  
  
random.seed()  
  
# Checks the function's behavior when given a length of 0  
**print** "gen\_sequence\_list(0, 2)"  
**print** gen\_sequence\_list(0, 2)  
  
# Checks the function's behavior when given a count of 0  
**print** "gen\_sequence\_list(2, 0)"  
**print** gen\_sequence\_list(2, 0)  
  
# Checks the function's behavior when given a length and a count of 0  
**print** "gen\_sequence\_list(0, 0)"  
**print** gen\_sequence\_list(0, 0)  
  
# Checks the function's handling of larger numbers  
**print** "gen\_sequence\_list(15, 3)"  
**print** gen\_sequence\_list(15, 3)  
  
# Checks the functions handling of negative numbers  
**print** "gen\_sequence\_list(-1, 5)"  
**print** gen\_sequence\_list(-1, 5)  
  
**print** "gen\_sequence\_list(5, -1)"  
**print** gen\_sequence\_list(5, -1)  
  
# Waits before running code that will crash the program  
raw\_input("Type anything to continue. ")  
  
# Checks the function's handling of floats (crashes)  
**print** "gen\_sequence\_list(5.0, 1.0)"  
**print** gen\_sequence\_list(5.0, 1.0)  
  
# Checks the function's handling of strings (crashes)  
**print** "gen\_sequence\_list(1, '5')"  
**print** gen\_sequence\_list(1, '5')

gen\_sequence\_list(0, 2)

['', '']

gen\_sequence\_list(2, 0)

[]

gen\_sequence\_list(0, 0)

[]

gen\_sequence\_list(15, 3)

['853908336192008', '816135274856394', '120427569230999']

gen\_sequence\_list(-1, 5)

['', '', '', '', '']

gen\_sequence\_list(5, -1)

[]

Type anything to continue. t

Traceback (most recent call last):

gen\_sequence\_list(5.0, 1.0)

File "C:/Users/SWK/Documents/.2A College Work Freshman/COS 125/Lab 3/lab3-3.py", line 44, in <module>

print gen\_sequence\_list(5.0, 1.0)

File "C:/Users/SWK/Documents/.2A College Work Freshman/COS 125/Lab 3/lab3-3.py", line 15, in gen\_sequence\_list

for i in range(count):

TypeError: range() integer end argument expected, got float.

Process finished with exit code 1

Problem 2:

**def seqzip**(seq1, seq2):  
 master\_list = []  
  
 min\_length = len(min(seq1, seq2))  
  
 **for** i **in** range(min\_length):  
 master\_list.append([seq1[i], seq2[i]])  
  
 **return** master\_list  
  
  
# Tests inputs of different lengths and types  
**print** "seqzip([1, 2, 3], 'abcd'"  
**print** seqzip([1, 2, 3], 'abcd')  
  
# Tests inputs of different lengths and types when one list is empty  
**print** "seqzip([], (1, 2))"  
**print** seqzip([], (1, 2))  
  
raw\_input("Type anything to continue. ")  
  
# Tests inputs of non-list types (crashes)  
**print** "seqzip(1, [])"  
**print** seqzip(1, [])

seqzip([1, 2, 3], 'abcd'

[[1, 'a'], [2, 'b'], [3, 'c']]

seqzip([], (1, 2))

[]

Type anything to continue. t

seqzip(1, [])

Traceback (most recent call last):

File "C:/Users/SWK/Documents/.2A College Work Freshman/COS 125/Lab 3/lab-3-3.2.py", line 24, in <module>

print seqzip(1, [])

File "C:/Users/SWK/Documents/.2A College Work Freshman/COS 125/Lab 3/lab-3-3.2.py", line 4, in seqzip

min\_length = len(min(seq1, seq2))

TypeError: object of type 'int' has no len()

Process finished with exit code 1