# Questions

### List Mysteries

1. Consider the following function:

def mystery1(lis):

 for i in range(0, len(lis), 2):

 element = lis[i]

 lis.pop(i)

 lis.append(element)

 print(lis)

Write the output produced by the function when passed the following list:

**List Output**

**(a)**

[2, 4, 6, 8] \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(b)**

[10, 20, 30, 40, 50, 60] \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(c)**

[-4, 16, 9, 1, 64, 25, 36, 4, 49] \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Consider the following function:

def mystery1(lis):

 for i in range(0, len(lis), 2):

 element = lis[i]

 lis.pop(i)

 lis.append(element)

 print(lis)

Write the output produced by the function when passed the following list:

**List Output**

**(a)**

[2, 4, 6, 8] \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(b)**

[10, 20, 30, 40, 50, 60] \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(c)**

[-4, 16, 9, 1, 64, 25, 36, 4, 49] \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Questions

### List Functions

3. Write a function remove\_even\_length that takes a list of strings as a parameter and that removes all of the strings of even length from the list. For example, if the list initially stores these values:

["to", "be", "or", "not", "to", "be", "hamlet"]

 After being passed to remove\_even\_length, it should store the following values:

["not"]

4. Write a function remove\_shorter\_strings that takes a list of strings as a parameter and that removes from each successive pair of values the shorter string in the pair. For example, suppose that a list called list contains the following values:

["four", "score", "and", "seven", "years", "ago"]

 In the first pair, "four" and "score", the shorter string is "four". In the second pair, "and" and "seven", the shorter string is "and". In the third pair, "years" and "ago", the shorter string is "ago". Therefore, the call remove\_shorter\_strings(list) should remove these, leaving the list as follows:

["score", "seven", "years"]

 If there is a tie (both strings have the same length), your function should remove the first string in the pair. If there is an odd number of strings in the list, the final value should be kept in the list.

5. Write a function named rotate\_right that accepts a list of integers as a parameter and rotates the values in the list to the right (i.e., forward in position) by one. Each element moves right by one, except the last element, which moves to the front. For example, if a variable named list refers to a list containing the values [3, 8, 19, 7], the call of rotate\_right(list) should modify it to store [7, 3, 8, 19]. A subsequent call of rotate\_right(list) would leave the list as follows: [19, 7, 3, 8]

6. Write a function named stretch that accepts a list of integers as a parameter and returns a new list twice as large as the original, replacing every integer from the original list with a pair of integers, each half the original. If a number in the original list is odd, then the first number in the new pair should be one higher than the second so that the sum equals the original number. For example, if a variable named list refers to an list storing the values [18, 7, 4, 24, 11], the call of stretch(list) should return a new list containing [9, 9, 4, 3, 2, 2, 12, 12, 6, 5]. (The number 18 is stretched into the pair 9, 9, the number 7 is stretched into 4, 3, the number 4 is stretched into 2, 2, the number 24 is stretched into 12, 12 and the number 11 is stretched into 6, 5.)

### Testing

**7.** A function called to\_single\_string that takes a list of strings as a parameter. It then adds up all the strings into a single string, with spaces in between, and returns it.

Example:

 a\_list = ["hi", "i", "am", "good"]

 ret\_str = to\_single\_string(a\_list)

 # This should result in ret\_str = "hi i am good"

What are some tests you could write to make sure the function works? (remember edge cases)

# Questions

**8.** The below code has some bugs. What would you do to try to find them? Write out the steps you take. Also, correct all style issues you find in the code.

def main():

 thefile = file\_to\_list(readme.txt)

 print\_everyother\_backwards(thefile)

#this function takes in a file name,

#then opens a file with that name,

#throws the contents into a list

#and returns that list

def file\_to\_list(filename):

 file = open("filename")

 the\_file = file.readline()

 the\_file = the\_file.split()

#this function reads in a list of strings

#and prints out every other line in the file

#backwords

def print\_everyother\_backwards(lines):

 print(lines[-1])

 for i in range(len(lines) + 1, 0):

 print(lines[i-1])

main()

# Solutions

[4, 6, 2, 8]

[20, 30, 50, 60, 40, 10]

[16, 9, 64, 25, 4, 49, 1, 36, -4]

[1, 2, 6, 8]

[10, 30, 40, 20, 60, 50]

[-4, 1, 25, 4, 16, 9, 64, 36, 49]

3.

def remove\_even\_length(list):
 i = 0
 while i < len(list):
 if len(list[i]) % 2 == 0:
 list.remove(i)
 else:
 i += 1

4.

def remove\_shorter\_strings(list):

 i = 0

 while i < len(list) - 1:
 first = list[i]
 second = list[i + 1]
 if len(first) <= len(second):
 list.remove(i)
 else:
 list.remove(i + 1)

 i += 1

5.

def rotate\_right(list):

 last = list[- 1]

 for j in range(list[-1], 0, -1):

 list[j] = list[j - 1]

 list[0] = last

6.

def stretch(list):

 result = [0] \* (2 \* len(list))

 for i in range(0, len(list)):

 result[2 \* i] = list[i] // 2 + list[i] % 2

 result[2 \* i + 1] = list[i] // 2

 return result