# More Function Practice

# Model Answers

#### **Challenge 1**

**Write a currency conversion program that accepts user entry to set the the conversion rate between New Zealand dollars and any other currency of your choosing. The program must then request a conversion type (either to or from) and the amount being converted. The program must do the conversion and display the result to the user before terminating.**

# MoreFunctionPracticeChallenge1.py

# Challenge 1

# Author: A. N. Other

# date: November 2016

rate = float(input("Please enter the conversion rate between NZD and GBP\n\n"))

type = input("Please enter 'to' or 'from'\n\n")

amount = float(input("Please enter the amount you wish to convert\n\n"))

def get\_convert(rate, type, amount):

result = float

if type == "to":

result = amount \* rate

else:

result = amount \* (1 / rate)

return result

exchange\_rate = get\_convert(rate, type, amount)

print("The amount of currency when converted {1} NZD is: {0} ".format(exchange\_rate, type))

'''

#assertions

Input: 0.44, to, 100

Output:

Please enter the conversion rate between NZD and GBP

0.44

Please enter 'to' or 'from'

to

Please enter the amount you wish to convert

100

The amount of currency when converted to NZD is: 44.0

#assertion2

Input: 0.44, from, 200

Output:

Please enter the conversion rate between NZD and GBP

0.44

Please enter 'to' or 'from'

from

Please enter the amount you wish to convert

200

The amount of currency when converted from NZD is: $454.54545454545456

'''

#### **Challenge 2**

**Write a program that takes a list of numbers (for example, list\_1 = [5, 10, 15, 20, 25]) and makes a new list of only the first and last elements of the given list. For practice, write this code inside a function.**

# MoreFunctionPracticeChallenge2.py

# Challenge 2

# Author: A. N. Other

# date: November 2016

list\_1 = [5, 10, 15, 20, 25]

def get\_first\_last(list\_1):

list\_2 = [list\_1[0], list\_1[-1]]

return list\_2

new\_list = get\_first\_last(list\_1)

print("The first and last items from the list are: ", new\_list)

'''

#assertions

Input: list\_1 = [5, 10, 15, 20, 25]

Output: The first and last items from the list are: [5, 25]

'''

#### **Challenge 3**

**Write a program that draws a game board that looks like this:**

**--- --- ---   
|   |   |   |   
 --- --- ---    
|   |   |   |   
 --- --- ---    
|   |   |   |**

# MoreFunctionPracticeChallenge3.py

# Challenge 3

# Author: A. N. Other

# date: November 2016

dashes = "---"

slashes = "| "

def make\_dashes(dashes):

print(dashes, end=" ")

def make\_slashes(slashes):

print(slashes, end=" ")

def board\_line():

for count in range (0, 3):

make\_dashes(dashes)

print("")

for count in range (0, 4):

make\_slashes(slashes)

print("")

for count in range (0,3):

board\_line()

'''

#assertion

Output:

--- --- ---

| | | |

--- --- ---

| | | |

--- --- ---

| | | |

'''

#### **Challenge 4**

**Define a function histogram() that takes a list of integers and prints a histogram to the screen. For example, histogram([4, 9, 7]) should print the following:**

**\*\*\*\*  
\*\*\*\*\*\*\*\*\*  
\*\*\*\*\*\*\***

# MoreFunctionPracticeChallenge4.py

# Challenge 4

# Author: A. N. Other

# date: November 2016

list\_1 = [4, 9, 7]

def histogram(list\_1):

for count in range (0,len(list\_1)):

for item\_count in range (0, list\_1[count]):

print("\*", end= " ")

print("")

histogram(list\_1)

'''

#assertion

Output:

\* \* \* \*

\* \* \* \* \* \* \* \* \*

\* \* \* \* \* \* \*

'''

#### **Challenge 5**

**Write a program that maps a list of words into a list of integers representing the lengths of the corresponding words.**

# MoreFunctionPracticeChallenge5.py

# Challenge 5

# Author: A. N. Other

# date: November 2016

list\_words = ["tiger", "leopard", "ocelot", "cheetah", "lion"]

list\_lengths = []

def word\_length(list\_words):

for count in range (0,len(list\_words)):

num = len(list\_words[count])

list\_lengths.append(num)

word\_length(list\_words)

print(list\_lengths)

'''

#assertions

list\_words = ["tiger", "leopard", "ocelot", "cheetah", "lion"]

list\_lengths = [5, 7, 6, 7, 4]

'''

#### **Challenge 6**

**"99 Bottles of Beer" is a traditional song in the United States and Canada. It is popular to sing on long trips, as it has a very repetitive format which is easy to memorize, and can take a long time to sing. The song's simple lyrics are as follows:**

**99 bottles of beer on the wall, 99 bottles of beer. Take one down, pass it around, 98 bottles of beer on the wall.**

**The same verse is repeated, each time with one fewer bottle. The song is completed when the singer or singers reach zero.**

**Write a program capable of generating all the verses of the song**

# MoreFunctionPracticeChallenge6.py

# Challenge 6

# Author: A. N. Other

# date: November 2016

start = 99

lyric\_1 = " bottles of beer on the wall, "

lyric\_2 = " bottles of beer. Take one down, pass it around, "

lyric\_3 = " bottles of beer on the wall."

def get\_full\_lyric(start):

print(start, lyric\_1, start, lyric\_2, start-1, lyric\_3)

start -= 1

return start

for count in range (0, start):

start = get\_full\_lyric(start)

'''

#assertion

Output: Increasingly less numbers of bottles of beer

'''