**COMP130 HW10: Lists  
instructor: John MacCormick**

Question 1. (5 points) Consider the definition of a list and the statements below:

days = ['Sun','Mon','Tues','Wed','Thur', 'Fri', 'Sat']

days[1:6]

days[2]

days[0]

days[5:]

days[:3]

Identify the statement from above that has each of the following values:

(a) 'Sun'

(b) 'Tues'

(c) ['Sun', 'Mon', 'Tues']

(d) ['Fri', 'Sat']

(e) ['Mon','Tues','Wed','Thur', 'Fri']

Question 2. (5 points) Consider the following definition of a list:

primes = [2, 3, 5, 7, 11, 13, 17, 23]

Give a Python expression using list indices and slices that has each of the values below.

(a) 7

(b) 23

(c) [3, 5, 7]

(d) [11, 13, 17, 23]

(e) [2, 3, 5]

Question 3. (8 points) Assume that a list named stuff has been defined. Write expressions that have the value of:

(a) The last element in stuff  
(b) The first half of the elements in stuff  
(c) The last half of the elements in stuff  
(d) stuff without the first and last elements

Hint: You'll need to use len

Question 4. (6 points) Consider the following list:

colors=['Red', 'Green', 'Blue', 'Yellow', 'Purple', 'Black', 'White']

Show what the colors list will look like after each of the statements below is executed. Assume that each statement operates on the list independently of the other statements – that is, the effects of the statements are not cumulative.

(a) colors[4] = 'Violet'

(b) colors[0] = 'Crimson'

(c) colors[1:3] = ['Emerald', 'Sky']

Question 5. (6 points) Consider the list below:

cars = ['volt', 'tesla', 'prius', 'highlander', 'camry', 'F150', 'mustang']

Write statements that perform each of the operations below. Assume that each statement operates on the list above independently of the other statements – that is, the effects of the statements are not cumulative.

(a) Replace 'camry' with 'corolla'  
(b) Replace both 'volt' and 'tesla' with 'electric' (electric should appear twice).  
(c) Replace tesla with ['3', 'X', 'S']

Question 6. (10 points)

Consider the following nested list, which stores information about course numbers (e.g. 130, 132, 232), section numbers (e.g. 1, 2) and instructors (e.g. Snape, Flitwick):

courses=[[130, [[1, 'Snape'], [2, 'Flitwick']]], [132, [1, 'Umbridge']], [232, [1, 'Trelawney']]]

(a) Write a statement that prints the number of different courses (not sections) being offered. Hint: Use len.

(b) Write a statement that prints the number of sections of 130. Hint: Use len on the list of sections nested in the list for 130.

(c) Write a statement that accesses and prints the name of the instructor for 232.

(d) Write a statement that changes the professor for 132 to be 'Hagrid'.

(e) Write a statement that changes the name of the instructor for section 2 of 130 to be 'Lupin'.

Question 7. (6 points) The concatenation (+) and replication (\*) operators that we used with string values also work with list values. Consider the lists shown below:

start=['a', 'b']

end=['x', 'y', 'z']

Use the above lists and the concatenation (+) and replication (\*) operators to give statements that will generate each of the following lists:

(a) ['a', 'b', 'x', 'y', 'z']  
(b) ['a', 'b', 'a', 'b', 'a', 'b']  
(c) ['x', 'y', 'z', 'a', 'b', 'x', 'y', 'z', 'a', 'b']  
(d) ['y', 'z', 'y', 'z', 'y', 'z']

Question 8. (10 points) Write statements using the variables defined below along with the append and extend methods to produce a list named elements that holds the first 10 chemical elements in order of atomic number (i.e. H, He, Li, Be, B, C, N, O, F, Ne).

elements = []

hydrogen = 'H'

helium = ['He']

two\_more = ['Li', 'Be']

six\_more = ['B', 'C', 'N', 'O', 'F', 'Ne']

Question 9. (8 points) Write a series of statements using insert to transform the initial value of the planets list given below into the full list of planets – so the final value of planets should be ['Mercury', 'Venus', 'Earth', 'Mars', 'Jupiter', 'Saturn', 'Uranus', 'Neptune']

The initial value is

planets=['Mercury', 'Earth', 'Mars', 'Uranus']

Question 10. (3 points) Write a statement that uses the remove method to remove Neptune from the list of planets constructed in your answer to the previous question.

Question 11. (6 points) Write statements that use the pop method to remove the planets on either side of Earth. Use the return value from pop to print the names of these planets.

Question 12. (10 points) In the dice game Yahtzee, a player rolls five six-sided dice on the first roll of each turn. Write a function named yahtzee\_roll that returns a list containing five values representing a roll of five six-sided dice.

Question 13. (9 points) Consider each task described below. Which pattern (map, filter or reduce) would be most appropriate for solving each task? Explain your reasoning briefly, in one sentence or less.

(a) Given a list of daily water depths in a river at high tide, find the mean water depth at high tide.  
(b) The Acme company gives all employees a bonus the holidays. This year it is a 15% bonus. Add the bonus into the scheduled pay for all of the employees.  
(c) A sheet metal manufacturing machine has a sensor that reports the thickness of the metal every 100 ms as it comes out of the machine. About once per second the sensor experiences an error and reports a negative reading. Remove the readings caused by errors from the data.

Question 14. (15 points) Consider the functions below. Which pattern (map, filter or reduce) best describes what the code is doing in each case? Explain your reasoning briefly, in one sentence or less.

(a)

def skip\_list(a\_list, jump):

new\_list = []

index = 0

while index < len(a\_list):

new\_list.append(a\_list[index])

index = index + jump

return new\_list

(b) Assume that the translate\_word function exists and has been imported from an appropriate external module.

def translate\_book(text):

translation = []

for word in text:

new\_word = translate\_word(word)

translation.append(new\_word)

return translation

(c)

def loop\_distance(points):

distance = 0

for index in range(len(points)-1):

pt\_dist = abs(points[index]-points[index+1])

distance = distance + pt\_dist

distance = distance + abs(points[len(points)-1]-points[0])

return distance

Question 15. (~~15 points~~ optional) Write a function named count\_words that takes two parameters: a list words and a string target. The function returns the number of times target appears in words. The case of the letters in the word should be ignored (forexample, Cat is counted as the same as cat). You can use the test function given below to check that your function is working correctly.

def test\_count\_words():

test\_list = ['a', 'B', 'c', 'C', 'b', 'b', 'd', 'b', 'd', 'e']

assert count\_words(test\_list, 'a') == 1, 'Wrong count for a'

assert count\_words(test\_list, 'b') == 4, 'Wrong count for b'

assert count\_words(test\_list, 'c') == 2, 'Wrong count for c'

assert count\_words(test\_list, 'd') == 2, 'Wrong count for d'

assert count\_words(test\_list, 'e') == 1, 'Wrong count for e'

assert count\_words(test\_list, 'f') == 0, 'Wrong count for f'

print("test\_count\_words succeeded")

Question 16. (~~15 points~~ optional) Write a function that given a list of words, generates a new list containing only the words that contain at least one double letter. For example, paddle would be included because it contains dd, but pop would not be included because even though it has two p’s, they are not adjacent. The case of the letters should be ignored. Hint: Use incremental development. Solve a simpler problem first. Write another function that solves the simpler problem and test it. Use this helper function as part of your solution to the original problem.

Total points on assignment: ~~137~~107