**Chapter 3 test problems**

1. Write a program that allows the user to input a list of names and then prints out the first and the last name of the list in lexicographic (dictionary) order.

>>> ========================= RESTART =========================

>>>  
Enter a list of names: ['Marcus', 'Fred', 'Carol', 'Alice', 'Bob']  
The first name is: Alice  
the last name is: Marcus  
>>>

2. Write a program that takes as input two opposite corners of a rectangle: the lower left-hand corner (x1,y1) and the upper right-hand corner (x2,y2). Finally, the user is prompted for the coordinates of a third point (x,y). The program should print Boolean value True or False based on whether the point (x,y) lies within the rectangle.

>>> ========================= RESTART =========================

>>>  
Enter x1: 1  
Enter y1: 3  
Enter x2: 10  
Enter y2: 6  
Enter x: 4  
Enter y: 4  
True  
>>> ========================= RESTART =========================  
>>>  
Enter x1: 1  
Enter y1: 3  
Enter x2: 10  
Enter y2: 6  
Enter x: 4  
Enter y: 2  
False

3. Suppose there is already a variable, called time, containing an integer representing the current time. Write code that prints “Good morning!” if time is less than 1200, and prints “Good afternoon!” otherwise.

4. Write a program that prints all the composers in the list

['Rzewski', 'Ruggles', 'Carter', 'Stockhausen', 'Stravinsky']

whose name has at least 10 characters.

>>> ========================= RESTART =========================

>>>  
Stockhausen  
Stravinsky

5. Write a program that prints all the composers in the list

['Antheil', 'Elgar', 'Saint-Saens', 'Chadwick', 'Coleridge-Taylor', 'Parker', 'Castelnuovo-Tedesco']

that do *not* have a double name. *Hint*: assume that a composer with a double name contains a space ' ' or a hyphen '-'.

>>> ========================= RESTART =========================

>>>  
Antheil

Elgar  
Chadwick

Parker

6. Write a function called tallEnough() that takes a single parameter, the user’s height in inches. If the height is 48 or more, the function should return “You can go on this ride.” Otherwise it should return the string “Sorry, you’re not tall enough.”

>>> tallEnough(46)

"Sorry, you're not tall enough."

>>> tallEnough(48)

'You can go on this ride.'

7. Write a function password\_check() that takes as input two strings newpassword and oldpassword, and that accepts the new password (i.e., **returns** True) if newpassword is different from oldpassword and newpassword is at least 6 letters long. If the new password fails the check, your functions should return False.

>>> password\_check('E10-s2ff', 'E10.s2ff')

True

>>> password\_check('E10sf', 'E10.s2ff')

False

>>> password\_check('E10-s2ff', 'E10-s2ff')

False

(Extra Credit) For extra credit, extend your password function so that it also checks that the new password contains a digit.

>>> password\_check('E10-s2ff', 'E10.s2ff')

True

>>> password\_check('Eej-sdff', 'E10.s2ff')

False

8. Write a function called average() that takes a single parameter, a list of numeric values. The function should return the average of the values in the list. If the list is empty, average() should return 0.

>>> lst = [1, 3, 4]

>>> average(lst)

2.6666666666666665

>>> average([])

0

11. Write a function called intersect() that takes 2 parameters, both lists. It then prints, one per line, all items in the first list that are also in the second.

>>> intersect([-3, 4, 76], [19, 27, 4, -3])

-3

4

>>> intersect([1, 4, 76], [19, 27, 5, -3])

12.

1. Write a function inside(x,y,x1,y1,x2,y2) that **returns** True or False depending on whether the point (x,y) lies in the rectangle with lower left corner (x1,y1) and upper right corner (x2,y2).

>>> inside(1,1,0,0,2,3)

True

>>> inside(-1,-1,0,0,2,3)

False

1. Use function inside() from part a. to write an expression that tests whether the point (1,1) lies in both of the following rectangles: one with lower left corner (0.3, 0.5) and upper right corner (1.1, 0.7) and the other with lower left corner (0.5, 0.2) and upper right corner (1.1, 2).

13. Implement function triangleArea(a,b,c) that takes as input the lengths of the 3 sides of a triangle and returns the area of the triangle. By Heron's formula, the area of a triangle with side lengths a, b, and c is , where .

>>> triangleArea(2,2,2)

1.7320508075688772

14. In written text small numbers are often written out, e.g. you'd write 'I have two brothers and one sister', rather than 'I have 2 brothers and 1 sister'. Here we want to implement one (1?) function that helps us realize this. Write a function smallnr() that takes a number x and if x is an integer between 0 and 6 it **returns** the name of the number, otherwise it simply returns x **as a string**.

>>> smallnr(2)

'two'

>>> smallnr(0)

'zero'

>>> smallnr(7)

'7'

15. Consider the following function:

def f(aList):

print(aList)

alist.append(3)

Suppose that function f() has been defined and that you are executing the below sequence of commands in the IDLE shell:

lst = [1]

f(lst)

f(lst)

f(lst)

What is printed when the code is executed?

Show the contents of lst after all the above code has run:

16. You can turn a word into pig-Latin using the following two rules (simplified):

* If the word starts with a consonant, move that letter to the end and append 'ay'. For example, 'happy' becomes 'appyhay' and 'pencil' becomes 'encilpay'.
* If the word starts with a vowel, simply append 'way' to the end of the word. For example, 'enter' becomes 'enterway' and 'other' becomes 'otherway' . For our purposes, there are 5 vowels: a, e, i, o, u (so we count y as a consonant).

Write a function pig() that takes a word (i.e., a string) as input and returns its pig-Latin form. Your function should still work if the input word contains upper case characters. Your output should always be lower case however.

>>> pig('happy')

'appyhay'

>>> pig('Enter')

'enterway'