**COS 120 - Introduction to Computational Problem Solving**

**Lab 07 – Lists and Dictionaries (100 points)**

**For the problems below, write the specified functions. Demonstrate they work correctly with appropriate test data. Demonstrate these to your lab TA's to receive credit.**

**Given:**

w=[83,99,2,3,1,7,54,1]

x=[23,12,67,5,4,11,2,84,12,16]

y={"staplers":2,"pencils":45,"erasers":12,"paper clips":200, "pens":84,"markers":12}

z={23012:2,77321:5,32332:234,77656:16,21321:802,99876:3}

**L07-01)** Write a function that will print out the contents of a list (1 point)

**L07-02)** Write a function that will print out key:value pairs of a dictionary as objects (1 point)

**L07-03)** Write a function that will print out key:value pairs (do not have to be objects) sorted by key (1 point)

**L07-04)** Write a function that will print out key:value pairs (do not have to be objects) sorted by value (1 point)

**L07-05)** Write a function that will prompt the user for a value and add it to a list (1 point)

**L07-06)**  Write a function that will prompt the user for a key and prompt the user for a value and then add the pair to a dictionary (1 point)

**L07-07)** Write a function that will prompt the user for a key and return the value for it found in a dictionary or return "No such value" if the key does not exist (5 points)

**L07-08)** Write a function that will prompt the user for a value and return true if it is in a **list** (**do so WITHOUT USING THE in OR not in OPERATORS!**) (5 points)

**L07-09)** Write a function that will return an **ascending sorted list** from an unsorted list passed to it as an argument (You may use any of a number of sorting algorithms, but **not the .sort() method**) (10 points)

**L07-10)** Write a function that will take two lists and merge them into one, maintaining their original **assumed** ascending sorted order (**do this WITHOUT resorting to the use of the sort method or sort function you just wrote, assuming the two lists passed to your function are already sorted**) (10 points)

**L07-11)**  Write a function to implement L07-10 any way you please! (5 points)

**L07-12)** Write a function to plot the following set of points on a graph. Search through the points to find the min and max x and y values and set the turtle world coordinates as (minx-10, miny-10, maxx+10, maxy+10). Draw appropriate x and y axes. Plot the points. Use the following list of lists to represent the points [[39,2],[16,5],[14, 99],[2,1],[28,12],[12,28],[20,50],[38,77]] (20 points)

**L07-13)** Write a function that accepts a dictionary as an argument. If the dictionary contains replicate values, return an empty dictionary, otherwise, return a new dictionary whose values are now the keys and whose keys are the values. (19 points)

>>> createNewValuesD({"a":5, "b":12, "k":13})

{12:"b",5:"a",13:"k"}

**L07-14)** Write a function that accepts a list of lists of homonyms and returns a dictionary that consists of each homonym as a key to a list of the rest of the homonyms, e.g. (20 points)

>>> homonyms=[["pie","pi"],["c","see","sea","si"]]

>>> hd=makeHomonymDictionary(homonyms)

>>> hd

{"pi":["pie"],"si":["c","see","sea"],"pie":["pi"],"sea":["c","see","si"],"c":["see","sea","si"],"see":["c","sea","si"]}