• Write all of your answers on separate sheets of paper. You can keep the exam questions when you leave. You may leave when finished.

• You must show enough work to justify your answers. Unless otherwise instructed, give exact answers, not approximations (e.g., $\sqrt{2}$, not 1.414).

• This exam has 6 problems. There are 240 points total.

Good luck!
Problem 1.
Write a python function that accepts a list of integers and returns a tuple containing the second smallest item in the list and the least index where the item occurs. (If all of the elements in the list are the same, return that number and index 0.)

Write your own search, don’t use the python list methods. (Ask me if in doubt.)
Test your function, of course.

Problem 2.
Write a python function that accepts a list of strings and determines how many times each item occurs in the list. The output should be a printed list of the items (in alphabetical order) and the number of times it occurs in the list.

Hint: Use a dictionary.

Problem 3.
Write a python function that accepts an arbitrary number of positional arguments and an arbitrary number of keyword arguments.

The output should be a printed list of the positional arguments and a table showing the keywords and the values assigned to them in alphabetical order by the keywords.

Problem 4.
Do problem 6 in section 13.6 in the book.

Problem 5.
This problem depends on some discussion in class. See me if you don’t understand it.

In this problem, we use a list of integers in the range 0, ..., n – 1 to specify a permutation of the numbers from 0 to n – 1. Write a function that accepts a permutation and prints out a tuple containing the following: a list of lists, where each sublist represents one of the cycles in the permutation (include the 1-cycles), and the sign (±1) of the permutation.
Problem 6.

This is a problem in object oriented programming. It’s a bit long to describe.

Implement a class Point that represents a point by storing a name and the x and y coordinates of the point. Include a method distance, so that if p1 and p2 are points, then p1.dist(p2) returns the distance between the points.

Implement a base class Shape that stores just a name and has methods

1. pt_is_in (a point)
2. area()
3. translate(vx,vy)
4. show_data()

In the Shape class, these methods should just return a message that the methods should be implemented by derived classes. The show_data() method should just return the name of the object.

Implement a subclass of Shape called Circle that specifies a circle by accepting the center (a Point) and the radius. The class Circle should implement the methods above for a circle. The translate method should translate the whole circle by the vector (vx, vy) (i.e., a point (x, y) is translated to (x + vx, y + vy)). The show_data() method should give the name and coordinates of the center point and the radius.

Implement similar classes for line segments and rectangles. The rectangles have their sides parallel to the coordinate axes and are specified by specifying two corners.

Create your own shape and design a class for it.

Test that all arguments make geometric sense and print error messages if they do not. Implement __str__ and __repr__ methods for all classes.

Test your code.