Recall the question on the first hour test that was diagnostic of HW2:

2. Your design of the floating point line segment intersection algorithm relies upon identifying the two special cases of when the lines are parallel and when they are collinear. Assume each line segment is represented by two endpoints, expressed as ordered pairs with floating point entries.

   a. Write pseudocode for these two conditions, carefully indicating if they need to be considered in any hierarchical order. (1 pt)
   
   b. If the line segments are determined to be collinear, do they intersect? Provide a brief proof or counterexample by picture to justify your answer. (1 pt)

The number of incorrect answers showed that this part of HW2 was not well understood. It is important towards the rest of culling, hidden line and hidden surface techniques that we will be building. Namely, this algorithm will typically be a pre-processor for those more advanced techniques.

Accordingly, this assignment will proceed to detailed design, implementation and testing of these concepts, as follows:

Design objectives: Take two line segments as input. Determine whether they are parallel and disjoint or parallel with some intersection. This should work for line segments of any slope.

Input:

1. Four ordered pairs, with each entry of each ordered pair in floating point arithmetic. Example input would be of the form

   \[ P_0 = (x_0, y_0), P_1 = (x_1, y_1), P_2 = (x_2, y_2), P_3 = (x_3, y_3), \]

   with each \( x_i \) and \( y_i \) being a floating point value, AND

2. A tolerance value used for floating point comparisons.

Output:

1. A text message with one of the following two possibilities;

   (a) parallel and disjoint, OR
   
   (b) parallel and intersecting.

2. If 1b, is true, then explicitly give the intersection set.

3. A listing of the input ordered pairs.

4. The tolerance value used for your floating point comparisons.
5. A graphics display of the two line segments, with each given different colors.

**Deliverables and Grading:** You should create, compile, debug and test your code in an environment that is both convenient for you and where you can demonstrate your results in real-time to me. Hence, if you’re working on a lap-top and can bring that to class, that should be fine. If you don’t have a lap-top, then we will make arrangements to reserve some SoE computers for your demonstrations, but it will be your individual responsibility to have pre-loaded and tested your software on those designated machines. At a designated time, I will post individual data sets for each of you to test, each set assigned to one individual. You will download and run the test data. I will examine your output and grade. **This will be discussed in detail in class on Tuesday, 2/28/06, so it is very important to attend.**