GiftWrap.zip contains code to randomly create points on a graph and display a simple polygon from those points. (This is already done for you)

From your output screen, the user should be able to select viewing options:
1. Points only: The anchor point labeled in black. All other points labeled in blue;
2. Simple Polygon: A simple polygon connecting the vertices colored in green.
4. Onion view: A series of convex hulls colored alternately in red or blue.

Create Simple Polygon (This part is already done for you. Study the code to make sure you know what was done.)

Hint: To create a simple polygon, you need the points ordered.

In any sort algorithm, we need a comparator function which compares values. All we have to do is have a “comparator” function for points.
We want D to be “greater than” C and “less than” E.
In that way, if we connect the points in terms of order, we can go clockwise around the set of points. Our orientation function can help us in determining which of two points is “greater”. Our anchor point serves as one of the three points we need in the orientation.

Part 1 Create Convex Hull
Implement the Gift Wrapping method (section 12.5.4) for determining a convex hull. Generate the code to display the convex hull.
Part 2 Create “onion” and test growth

An onion of a set of points is a series of convex hulls formed by finding a regular convex hull, throwing away the points involved, and finding the next convex hull. This is repeated until no more convex hulls are possible. The hypothesis is that the number of layers of the onions grow like $c \cdot n^e$

Compute the onion of $n$ points drawn in a square (pick each coordinate in [0,600] randomly) and count the number of layers. Let $L_n$ be the average number of layers. Run your code for many values of $n < 10^6$ and estimate the growth of $L_n$ as a function of $n$.

The goal of this exercise is to test the hypothesis that $L_n$ grows like $c \cdot n^e$ for some constants $c, e > 0$. Experiment with many point sets to guess a plausible value of $e$.

Once the points have been entered, the system will display a series of convex hulls and report the number of layers.

**Hints**  For the y axis, down is larger. I found it easier to think of the y coordinate as negative. (It is like I viewed the graph in the fourth quadrant.)

**Report (turn in as readme.txt)**

How many hours did this assignment take?
How does the number of layers grow with the number of points? Show a table of results to justify your answer.