Homework Assignment 4

Given 04/28/2016, due 05/16/2016

Write code that finds the longest x-monotone path in an arrangement of lines. The lines are given by their equations $\texttt{a[i]}x+\texttt{b[i]}y = \texttt{c[i]}$ for $\texttt{i} = 0, \ldots, \texttt{n} - 1$. The path starts at an intersection point, and follows the lines, possibly switching on intersection points to different lines, but always going from left to right, until it ends at an intersection point. You create a function

```
void longest_path(int n, int *a, int *b, int *c)
```

The function prints the sequence of lines taken by the longest path, and the total length.

In the function, you first compute all intersection points of all lines (there will not be any parallel lines). You use a structure

```
struct point { double x; double y;
               int line1; int line2; double length }
```

to store the intersection point of lines i and j; `x, y` are the coordinates, `line1, line2` are i and j, and `length` is the length of the longest x-monotone path that ends in this point. The intersection of lines $a_1x + b_1y = c_1$ and $a_2x + b_2y = c_2$ is given by $x = \frac{b_2c_1-b_1c_2}{a_1b_2-a_2b_1}$ and $y = \frac{a_1c_2-a_2c_1}{a_1b_2-a_2b_1}$.

You first generate all intersection points, each point labeled by the two lines on which it lies, and initially with `length` $= 0$. Then you sort the points with increasing x-coordinate. Then you process the points in sequence of increasing x-coordinate (left to right). If point p lies on line i and j, and on line i it is not the first point, but point q is the next point to the left of p on line i (scan array of points before p from p backwards to find q, if it exists), if $\texttt{q.length} + \texttt{dist(p, q)} > \texttt{p.length}$, you increase `p.length` to that value. The same for line j. After these two possible updates for `p.length`, you are finished with `p` and move on to the next point.

Then, after all points received their `length` value, you make a second pass to find the largest length, and work from it backward to reconstruct that longest path.