**Part 1:**

*Description: This method is used to decide whether or not two points can see each other "over" a wall. This first calculates the y-value of the intersection of the two points with the wall. Then the method makes sure the points are not on the same side of the wall. Check if the points are*

*Input: Two arrays (the two points) and one matrix (the wall)*

*Output: True if the points can see each other, false otherwise*

Take in points, and instantiate integer lists like so:

L = [X,Y]; R = [X, Y]; Wall = [[X,Y],[X,Y]]

wallYIntersect = Ly + (Ry - LY)\*(WX - LX) / (Rx - LX)

If LX and RX less than WallX

Return true

fi

If LX and RX greater than WallX

Return true

fi

Else if either LY or RY are less than wallYIntersect and the other is greater

Return false

fi esle

Else

Return true

fi esle

**Part 2:**

*Description: This method is used to decide whether or points from a matrix can see each other "over" a wall. The method loops through the given matrix of points and applies the method created from part 1 to decide if the points intersect the wall. If the points can "see" each other then the method adds them to a matrix that is then printed out for the user to see*

*Input: One matrix of the points and one matrix (the wall)*

*Output: Matrix of points that do not intersect the wall*

int count starts at 0

For 0 to pointMatrix

If Part1 returns true for the point

Increase count by one

fi

rof

Instantiate outputMatrix of size count

For loop through pointMatrix

If Part1 returns true for the point

Add point to outputMatrix

fi

rof

print out outputMatrix