Quiz 9: CS4640 Name ____________________

1. For each of the lines below, give its Cartesian (i.e., slope-intercept form) equation and its standard form (i.e., a,b,c coefficients).

1a.

```
   ^                             Cartesian          Standard
y  |   ^
   |   |                           m = not            a = 1
   |   |                           b = possible       b = 0
   |   |                                              c = -1
   +---o---------------->
       |[1;0]            x
       |
       |
       V
```

1b.

```
     ^                             Cartesian                Standard
  y  |
     |
 <---o----->                         m = 0                    a = 0
[0;1]|                               b = 1                    b = 1
     |                                                        c = -1
     +------------------>
                   x
```

1c.

```
      ^ y                              Cartesian           Standard
    ^ |
     \|
      \ <---- theta = 135 degrees        m = -1              a = sqrt(2)/2
      |\                                 b =  2              b = sqrt(2)/2
      | o[1;1]                                               c = -sqrt(2)
      +--\---------------->
          \         x
           v
```

1d.

```
   ^                                   Cartesian           Standard
y  |
   |
   |  ^                                  m = 1               a = -sqrt(2)/2
  1| /                                   b = 0               b = sqrt(2)/2
   |/  <----  theta = 45 degrees                             c = 0
   o------------------>
  / [0;0]           x
 /
V
```

2. Give a robust (e.g., consider the parallel relation) Matlab function to compute the smaller angle between two lines as indicated in the function header. Recall that lines may be parallel (and either completely intersect or don't at all) or they intersect in a point.

```
function theta = CS4640_lines_angle(a1,b1,c1,a2,b2,c2)
% CS4640_lines_angle - find smaller angle between two lines
% On input:
%     a1 (double): x coefficient of first line
%     b1 (double): y coefficient of first line
%     c1 (double): constant coefficient of first line
%     a2 (double): x coefficient of second line
%     b2 (double): y coefficient of second line
%     c2 (double): constant coefficient of second line
% On output:
%     theta (double): smaller angle between lines
% Call:
%     t = CS4640_lines_angle(1,0,-1,0,1,1);
% Author:
%     <Quiz Taker>
%     UU
%     Spring 2018
%

dir1 = [a1,b1];
dir1 = dir1/norm(dir1);
dir2 = [a2,b2];
dir2 = dir2/norm(dir2);
theta = posori(acos(dot(dir1,dir)/(norm(dir1)*norm(dir2))));
if theta>pi/2
    theta = pi - theta;
end
```

3. Give a detailed explanation (algorithm) of how straight lines can be found by using the gradient to produce the basic info, and then combining pixels to produce line segments. Assume the input is a binary image and the output is an image in which each line segment has a unique label. Discuss the strengths and weaknesses of your method.

```
Get mag and ori from dx and dy

Visit every pixel, p
  if p has no label and mag(p) above threshold
    set param as ori(p)
    increment label
    add p to  OPEN
    while OPEN~=empty
      p' <-- Pop(OPEN)
      if ori(p') close enough to param and p' not labeled
                 and mag(p') high enough
        label p' with label
        add p' neighbors to OPEN if they are not labeled, have
                ori close to param and have large enough mag
      end
    end
  end
end
```