# Pseudo code

## Variables:

vLeft: Speed of left motor – int

vRight: Speed of right motor – int

vForward: Constant speed of robot to go forward – int – 20

vDifference: Difference between motor speeds – int – not initialized

row: Row number – int – 50

error: Distance of line from center – int – not initialized

pixArray: Array to store luminance of pixels – int – size 150

## Functions:

isWhite(int pixelValue)

* Receives a number containing luminance value of a pixel, ranging from 0-255
* Returns an int
* Checks if the pixel is white (threshold is if value > 250)
* If white, return 1
* Else, return 0
* End function

findError(int arrayCenter, int lineCenter)

* Receives two numbers, the center of the array, and the center of white line in array
* Returns an int
* Finds the difference between the array center and the white line center
* And returns this value
* End function

(Continued on next page…)

main()

* Check if robot exists
* Initialize variables
* Main while loop for robot
  + Take picture and save as PPM file. Do not change name of PPM file.
  + For all columns in a fixed row,
    - Get luminance of pixel
    - Convert this to 1 if it is white, 0 if not white
    - Store this in the array of pixel values
    - End for
  + Calculate average value of the indexes of white pixels in the array – this value is the center of the white line
  + Compare this to actual center of pixel values array
  + Calculate the error – distance of line from center
  + Calculate difference in speed of left and right motors
    - Difference = k \* error
  + Calculate the right and left speeds of the motor according to error.
  + Set speeds of motors to vForward +/- vDifference for right/left motor
  + End while
* End main

## To add:

Lost mode

* What to do if robot loses the line?