WTF DOES THIS PROGRAM DO YOU SAY?

- GUI
    - Insert the blank scantron image
    - Insert the student scantrons along with the key
    - Slap that grade button
    - Ports all the scantron information to separate .csv files
        - First Name
        - Last Name
        - University ID
        - Additional Information
        - All the correct key answers
        - All the student answers
        - The student score
- Main Code SHIT (“Takes 4 shots of vodka”.... LETS GET IT)
    - Make a Mat to hold the blank scan sheet image
    - Make a 2D Mat to hold however many images the user inputs
    - Convert the blank scan sheet image to grayscale
    - Apply Hough Circles to the blank scan sheet image
        - This is to get a circle location for each bubble on the scantron
    - The scantron has 1450 bubbles but Hough Circles detects 1457!
        - O NO…. WHATEVER WILL WE DOOOOOO
    - Fuck you extra random circles…… “7 other circles have left the vector”
        -): “sniff sniff” said the circle
            - Fuck off
    - Now, let's back up a little and think this through
    - The circles are stored in a vector<cv::Vec3f>
        - Each circle has an x, y, and radius. (x,y,radius)
    - We can take the mode of the vector in terms of the radius alone
        - All 1450 bubbles have the same radius of 30.5 on a 500 DPI image
    - We can then disclude all of the circles that do not have this radius
        - This is how we remove those 7 circles!
    - Since the radius is constant for all of the circles lets just store them in a vector<cv::Point> so it is less complex
    - Now we have 1450 circles x,y coordinates which mark the locations of the bubbles on the blank scantron
    - But what is this??? Hough Circles does not order the circles from top to bottom and left to right??
    - We can use std::sort and a custom compare function I made to do this!
    - Now them bubbles are nice and organized, but NOT GOOD ENOUGH
    - We wanna split the circles up into categories (First Name circles, Last Name circles… etc.)
    - Call my magick circle organizer function to organize all of the circles

- BOOM. Circles are organized perfectly into categories
- Each category (First Names, Last Names…) gets its own vector<vector<cv::Point>>
- “And it was at this moment he knew, he fucked up”
- I ORGANIZED SOME OF THE CATEGORIES WRONG…. FUCK ME
- Scurries and fixes that shit real quick
- With a little flick of his wrist he transposes the fuck out of some multidimensional vectors to properly organize everything
- Alright, everything is now perfectly organized and we can relax
- Vlad walks in…. TIME TO DO SOME MAPPING
  - Fuck off Vlad….
- Converts each of the student scantrons to grayscale
- Add some gaussian blur
  - Why?
    - Because Vlad does whatever the fuck he wants
      - Leave him alone
- WTF, the blank scantron sheet has different dimensions than the student sheets
- O shit, there are targets on these student scantron sheets
- Wait, there are also targets on the blank scantron sheet
- Let’s use Hough on both the blank scantron sheet and the student scantron sheets
- Now we have the target locations of both sheets!
- Vlad whips out his hand and performs a perspective transform to match the student scantron dimensions to the blank scantron dimensions using the targets on both scan sheets
- But wait! What is that?
  - “Carl, I don’t feel so good”
    - Carl snaps his fingers
      - 50% OF THE SCANTRONS NOW HAVE NON UNIFORM SLANT
- Carl does not know how to scan a scantron?
  - Oof
- Meanwhile…. Vlad sends me broken code that slants the student scantron more than Carl’s hairline
- Reverts back to Vlad’s old code that works like a charm
- Alright, let’s get back on track
- Both sheets have the same dimensions! YEET
- Let’s do some image pre-processing on the student scan sheets to make the filled bubbles darker, bigger, thicker, and (You have a dirty fucking mind….)
- Firstly apply inverse global thresholding to binarize the student scantron
  - The threshold value T will be determined from 255 minus the standard deviation of the image
    - Why?

      - Because it fucking works nice
- Then let’s erode the image to get rid of any stray marks and thin the bubble thickness
- Then apply dilation to expand any student filled responses that were small
- Re-invert the student scantron because we are done with morphology
- Now let’s resize the student scantron to 25% of its original size
  - Why?
    - SPEEEEDDDD
- Now let’s grade these inbred student scantrons
- Recall from earlier that all of the blank scantron circles are stored away in a bunch of nicely organized vectors
- Also recall that the student scantron was perfectly mapped to the blank scantron
- But we did resize the student scantrons to 25% of their original size
- We can just scale the circle coordinates down to that same % and they will fall right on top of the bubbles on the student scantrons :D
- Now we look at each bubble on the student scantron
- For each bubble, if the mean in the bubble is less than 100 we have a match
- We store all of the First Name, Last Names, answers…… in vectors
  - Jeez, it is 2AM, I have been typing for 15 minutes now….
- We will also have a 2D vector to store each of the scantron responses
- Once we have all of the scantron responses we must search for the KEY!
- Just loop through all of the Last Names real quick and the last name that is equal to KEY is probably our key lol
- However, we just want to extract the answers from the key into a new vector called correct answers
- We then compare the correct answers against all of the other student scantrons
- Wait, how do we know how many questions long the exam is????
- Just count the number of questions in the KEY exam duh….
- At this point we basically have everything that we need
- We can begin piping stuff out to a .csv file
- Clear all memory and we are done….