One of the design decisions I made was to have a seperate file of helper functions that I could use throughout each slice of the assignment. I had a function that checked if a file existed, logged commands, got a filename, and created a new raster Image struct. The function that created the new raster image struct proved to be most useful, As I could use it to read, create a raster, and create a 2d raster all with one line of code.

All of my programs except for dimensions dont have error handling for command line arguments. Someone could enter random stuff for a few of them and they would still run. For example with my rotation program, to see if there is a double rotation I just check to see if the second arguments string length is equal to 2.

There were a few problems that made this assignment a lot more tedious than it should have been.Many of the specifications were a little ambiguous, so I had to figure things out by surfing the forums constantly. For example, for the rotation part of the program, the instructions didn't say where the starting point of the crop would be. I wasn't sure if it was at the top left or in the middle of the square. So I just chose top left. Another problem was that for the bash script you were supposed to have something that split the picture into 4 quadrants. The only problem with that is that if you didn't do the extra credit (adding the number to the end like so [cropped 1] [cropped 2], this part wasnt possible. The reason is that everytime the crop runs with the same input the output will just overwrite the previous run. So even if you run ./crop 4 times you will only have the cropped photo of the last command.

For the extra credit I used doxygen and implemented the 2d raster.