Your homework this week will consist of generating the image files below and answering the questions. Please name the images according to the format: Lab1\_software\_yourname\_problemNumber.format (for example: Lab1\_MATLAB\_sara\_1.tif) Write the answers to the questions in a text document: Lab1\_yourname\_answers

Homework is due next Thursday, email everything in a single email to sara@ucsc.edu.

Thus, you have one more Lab class to work out and resolve any problems or difficulties!

* Load the mushroom image from the course website.
  + In MATLAB you can use the command “**imread()**” to read a file. If you want help to understand how this command works, type **help imread** in the command window
  + This image is a grayscale image but saved as a .jpg. Therefore, it will open as an x,y,z array where xy is the number of pixels of the image and z=3 which is the number of matrices. If you inspect these elements you will see that all three are the same. Therefore, take only one of these three 2D matrices and work with. For example like this:   
    M = imread(‘filename.jpg’);  
    Mbw = M(:,:,1);

Now Mbw is the grayscale 2D matrix containing the black and white image

* Display the image in a grayscale lookup table
  + You can use e.g. the command **imagesc** to plot the image. If you use this command, to plot the image without distortion in MATLAB, use the commands “axis equal” and “axis tight”)
  + If you want to see how this works, you can also try displaying the image in a false lookup table. The MATLAB command for these things are:  
    colormap gray  
    colormap hot  
    … and many other colormaps.
* Display duplicate copies of the image and adjust the contrast in these so that you can see different parts of the image on the monitor, such as:
  + The cap
  + The gills
  + The structure of the table the mushroom is laying on
  + Invert the image lookup table of one of these images
    - In MATLAB you can use the command imagesc(M,[min max]) to display the image between the grayscale values min and max. Experiment with some different values!

1. Plot these four images in a single window and **save** as an image file (choose a format).  
   **Question**: How many pixels does the image have (expressed as *x* times *y*)?

* Mirror (flip) the image
* Rotate the image 90 degrees
* Add a scale bar to the (original) image. (We don’t know the size so make something up.)

For MATLAB: also add legends to the axes (for example, label them x and y)

* Binarize the image. (Extra: Explore different thresholding functions, e.g. renyi)

1. Plot these four images in a single window and **save**   
   **Question**: what is the bit depth of the original image?
2. Plot the intensity of a line drawn through the mushroom. (For MATLAB how to, see Example 2.5 in *DIP using matlab*). **Save** as an image.
3. **Write and save a script or macro** that does all the tasks in **1**.
4. MATLAB ONLY: CHECK OUT WHAT HAPPENS WITH JPG IMAGE COMPRESSION  
   In MATLAB: Follow the book (Digital Image Processing in MATLAB, Gonzales and Wood, 2nd edition) example in 2.4 for compressing jpg images. Try this with the mushroom. When do you start losing detail and information? How much difference does this do in terms of file size? **Question:** Write a couple of short sentences describing what you find.
5. MATLAB ONLY: CHECK OUT THE MATLAB DATA CLASSES uint8, uint16, double, logical. Familiarize yourself with these. Convert the image back and forth between them. Does the image change when you do that? **Question:** Some conversions are not lossless/reversible, but you lose information. Give an example of this.
6. MATLAB ONLY: CHECK OUT THE MATLAB MATRIX INDEXING (2.8.2) work through the examples to learn how to manipulate matrices in MATLAB. Write and **save** two script that spit out the numbers 3-7. Write one script using a *for* loop and one that uses a *while* loop.

* Since this turned out to be a confusing question, here is an example that does this:

x = 1:10

for n = 3:7  
 x(n)

end

Now just write a loop that does the same thing using a *while* statement!