CS 420 – Advanced Programming Languages

Assignment 5 – Scripting in Python – 100 points

In this assignment, you will be writing a simple imperative program in C, functional program in Haskell and a logical program in prolog. You will leverage the image processing capabilities of MATLAB and write a script file on python to integrate all the codes written.

Objective:

In this assignment you will

1. Write a MATLAB script to read an input image and process it.
2. Write a C program to apply thresholding operation to obtain a black and white image.
3. Write a Haskell Program to invert the colors of an image.
4. Write a Prolog code to rotate the image.
5. Write a MATLAB script to display all the generated images.
6. Write a wrapper python script to integrate all the written codes and seamless transition between various programming paradigms.

Submissions:

1. You will turn in two files on to canvas.
    1. A zip file containing all the codes. The python code should be called wrapper.py
        a. All your codes should have detailed comments. This will help you to refer to the project before an interview.
    2. A simple report including screenshots/proof that your codes work.
        a. Add a screenshot of your output from day 1.
        b. Add screenshots from day 2 to illustrate that your C, Haskell and prolog codes work as desired.
        c. Indicate progress. Were you able to complete the lab?
        d. Output images from the program.
            i. Show the output for provided mickey.png file.
            ii. Rerun the program with any image of your choice.
        e. Acknowledge the references you used.

Task 0: Installing the required software

1. gcc for C
2. ghc for Haskell
3. swipl for Prolog
4. MATLAB
5. python3

Note: all the tools should be accessible via the command line

Task 1: C program

1. Write a C program to read a sequence of number from the command line.
2. Dynamic memory allocation should be leveraged to store the incoming values to a variable.
3. Iterate though the stored array.
    a. If the value is greater than threshold – print 255
    b. If the value is lower than threshold – print 0

Note: The optimal value of threshold for the provided mickey image is 170. Figure out the threshold value using trial and error for other images.

Task 2: Haskell Program

1. Write a Haskell function that would subtract the elements in provided list from 255.
2. Haskell program should integrate the IO( ) functionality to allow command line execution of the code.

Task 3: Prolog Program

1. Write a prolog rule to reverse a list.
2. Write a wrapper rule called main that would read inputs from command line (std_in), call the reverse_list rule and output the result on command line (std_out)

Task 4: MATLAB Program 1

1. Write a MATLAB program to read and process a greyscale image.
    a. The code should first get the image pixel values represented as a matrix.
    b. Then convert the matrix to a 1D array.
    c. Store the 1D array in a text file. Call the file ‘input.txt’.

Task 5: MATLAB Program 2

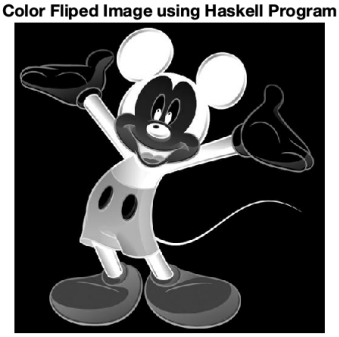
1. Write a MATLAB program to read the results of C, Haskell and Prolog codes stored in text files.
    a. The code should read the values as 1D array.
    b. The array should be treated as pixel values – unsigned chars.
    c. Reshape the array back to a 2D matrix.
    d. Visualize the matrix as images and plot the images.

Here is a sample output.

Original Image

Black & White Image using C Program

Color Fliped Image using Haskell Program



Rotated Image using Prolog Program

Task 6: Python wrapper program

1. Python code should first run the MATLAB program 1 to generate the input array.
2. Your code should use file read operation should read the array from the text file ‘input.txt’.
3. Call the C Program to perform thresholding.
    a. Save the output in a text file. Call the file ‘c_output.txt’.
4. Call the Haskell Program to perform color flip.
    a. Save the output in a text file. Call the file ‘haskell_output.txt’.
5. Call the C Program to perform rotation.
    a. Save the output in a text file. Call the file ‘prolog_output.txt’.
6. Finally call the MATLAB program 2 to read from the three output files, reconstruct images and plot them.
    a. You may use subplot feature to plot all the images in a single image.
    b. Add titles to your plot.
    c. Explore other plotting options available to make your output stand out.

Task 7:

- Include this project on your CV immediately. Your recruiters will be intrigued by this project and will likely engage in discussions with you about it.
- Jot down the tasks you accomplished (minor details in the code) for future reference before your interview.

☺