Assignment-3

Jason D1228792 4/10

When doing the homework, I have met some problems.

I don’t know how to put the image into the code, so I ask Jack and finally hot the meth.

I don’t know why the merged image I made was upside down but left and right was not reversed. Finally, I realized that the beginning bmp image was started from left down corner.

program steps:

First, I made a type defined for the image data. Then, I made two functions. The first one named Write_data was for writing the data I calculated in the code to the file. The second one named print_data was for printing the information of the image data. Next, I set up some variables and open the file with ‘fopen’. Also, writed the information of the image into header. Then, reduced the image into reduce_image. Finally, I made the merge image by for loops and free the space of the array.

For the steps of making the merge image.

The code begins with nested loops iterating over the height and width of an image represented by the heafin object.

Inside the loops, it calculates the index kfin to access pixel data in the final image (finimageData) based on the current coordinates (i and k).

It then checks conditions to determine which part of the image frame the current pixel belongs to:

If it's within the vertical frame, horizontal frame, or the four corners, it sets the pixel data to predefined colors (fcol1, fcol2, fcol3) accordingly.

If it's inside the main image area (not part of the frame), it copies the corresponding pixel data from another image (reimageData) with some adjustments based on the coordinates and sizes.

The conditions account for different regions of the image, such as the frame edges and corners, as well as the main image area.

Finally, it loops through all pixels, processing each one according to its position relative to the frame and main image area.

Overall, the code seems to be creating a new image with a frame around it, possibly for visual enhancement or manipulation purposes.