**DESIGN DOCUMENT**

**PEYTON OTEY**

**IMAGE PROCESSING**

DATA:

1. Image Data:
   1. Field: 2D array of integers (8 x 8) representing the pixel values of a loaded image.
2. File Pointers:
   1. A way to access the image file.
3. Image Editing State:
   1. 2D array of integers representing the edited image back to the user.
4. User Interaction:
   1. Loops and switches: To decide between choices and if the user would like to further edit the image or exit.
5. Saved Image:
   1. 2D array of integers that represents the final image after all edits

PROCESS (Algorithm):

* Prompt the user with a menu that can do one of the following:
  + Load a new image
  + Display the current image
  + Edit the current image
  + Exit the program
* Decide which selections the user wants and from that do the performed task.
* Return the newly edited image back to the user and prompt for more options.
* Continue to edit the image till the user selects to exit out of the program.

FUNCTIONS:

1. Main Function:
   1. Provide the user with a menu of all the selections. Depending on the user input or choice, call the specific function to execute the needed process and return the user with the desired outcome.
2. Load\_image (filename):
   1. Input Parameters:
      1. ‘filename’(string): The name of the image file that’s going to be loaded
   2. Returned Output:
      1. Image data.
   3. Functionality:
      1. Reads the contents of the image file provided by the user.
3. Display\_image (Image\_data):
   1. Input Parameters:
      1. ‘Image\_data’ (2D array): The data representing the image to be displayed
   2. Returned Output:
      1. None.
   3. Functionality:
      1. Displays the image that is represented by the ‘Image\_data’ parameter to the user.
4. Edit\_image (Image\_data):
   1. Input Parameters:
      1. ‘Image\_data’ (2D array): The image that needs to be edited in any way.
   2. Returned Output:
      1. Edited image data (2D array).
   3. Functionality:
      1. Provides a menu that the user can select an option. These being cropping, dimming, brightening, or rotating the image provided.
      2. Performs the selected edit on the ‘Image\_data’ (2D array).
      3. Returns the now edited image back to the user to then be decided for further instruction.
5. Crop\_image (Image\_data, start\_pixel, end\_pixel):
   1. Input Parameters:
      1. ‘Image\_data’ (2D array): The image that needs to be cropped.
      2. ‘Start\_pixel’, ‘End\_pixel’ (pixels): The area within the pixels that needs to be cropped..
   2. Returned Output:
      1. Cropped image data (2D array).
   3. Functionality:
      1. Takes out the specific spaces the user has inputted.
      2. Returns the cropped portion of the image back to the user.
6. Dim\_image (Image\_data):
   1. Input Parameters:
      1. ‘Image\_data’ (2D array): The image that needs to be dimmed.
   2. Returned Output:
      1. Dimmed image (2D array).
   3. Functionality:
      1. Reduces the brightness of each pixel that is within the image.
7. Brighten\_image (Image\_data):
   1. Input Parameters:
      1. ‘Image\_data’ (2D array): The image that needs to be brightened.
   2. Returned Output:
      1. Brightened image data (2D array).
   3. Functionality:
      1. Increases the brightness of each pixel that is within the image provided by the user.