ABDUR RAHMAN
04072213016
BS 6TH SEMESTER

1. **Problem Definition**
   The given assignment requires the completion of tasks such as reading/writing **.ppm** binary files, applying filters and color combination, subtracting two images, adding two images and generating a sequence of transitioning **.ppm** images.

2. **Input Data**
   Binary ppm image files are input to the program, user gives the name of input file, program reads that file and the output name of file is also given by the user, and program uses that name to generate the output file.

3. **Data structures**
   I used my own customized classes such as ***PPMImage*** and ***PGMImage*** to store and process the input and generate output. The class contains *Red*, *Green*, and *Blue* unsigned char* to dynamically allocate memory and contains *height*, *width*, *magicNumber* and *maxVal*.

   - ***readPPM(<filename>)*:** to read the contents of file using the filename passed as argument.
   - ***writePPM(<filename>)*:** to write the processed contents of file using the filename passed in the argument.
   - ***invertColors()*:** to make the negative of the file by subtracting the RGB channels from the max value i.e. 255.
   - ***makeRed()*:** make the image red by making the Green and Blue channel 0.
     - *Similarly makeGreen() and makeBlue().*
   - ***makeYellow()*:** makes images yellow by making green and red equal and blue to 0.
     - *Similarly makeCyan(), makePurple().*
   - ***minus(<obj>)*:** Subtracts the image passed in the argument from the calling image object subtracting their RGB channels.
   - ***plus(<obj>)*:** Adds the two images by adding their RGB channels.
   - ***makeTransitions(<obj>)*:** generates 10 images transitioning from one image to the other by using the parametric equation of line.
   - ***makeGrey()*:** generates greyscale image by averaging the RGB channels and make new image with ***.pgm*** format.

4. **Output**

   Output images generated are in .ppm format and .pgm format.

   Steps to generate ppm images:

   - User enter filename.
   - Program read the file and asks user to pick a color to convert the file into

- Program tweaks the color channels such as making Red and Blue to 0 if user want Green image and similarly make Green and Red to equal and making Blue to 0 to generate Yellow image, or even invert the color channel by subtracting from 255 and so on.
- User can also give two filenames and can pick operations to perform on those files.
- Program read two files in memory and perform addition or subtraction on those files, new channel created after addition or subtraction remains in the 0 to 255 range.
- User can also make transitions by choosing two images and program generates 10 images making rough transitions.

Steps to generate pgm images:

- User gives the filename and selects the greyscale option.
- Program reads the input file and average the RGB channel (R+G+B)/3 generate the greyscale image, keeping resolution same but changing the extension to pgm and magic number to p5.

5. Challenges

- Reading a binary ppm file is a bit of a challenge due to lack of practice and took time to refresh the concepts of filing and typecasting pointer while reading/writing in binary format.
- Working with classes and manually handling memory and pointers is a bit tricky but exciting. (nostalgia hit me hard while getting memory leaks exceptions! 😭).

6. Sources

i. *CppNuts from YouTube, for understanding reinterpretcast and binary file handling.*
ii. *OpenAI for understanding the errors related to memory leakage.*
iii. *My past programs on Object-Oriented programming in C++.*