2a. Provide a written response or audio narration in your video that: w identifies the programming language; w identifies the purpose of your program; and w explains what the video illustrates. (Must not exceed 150 words)

2b. Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and/ or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development. (Must not exceed 200 words)

My program began with the problem of what my idea would be. I had an idea in mind, however, I realized the hard way that it would be difficult and would require a graphical user interface. So, Mr. Orkney (my CSP teacher) who knew of my Scarlett Johansson obsession in his class asked, “Wouldn’t it be better if Scarlett was a little more orange?”. This gave me the idea of creating an orange filter where the user can insert any picture of their choosing and it would be immediately changed to an orange color scheme based off of the amount of the addition of RGB values.

2c. Capture and paste a program code segment that implements an algorithm (marked with an oval in section 3 below) and that is fundamental for your program to achieve its intended purpose. This code segment must be an algorithm you developed individually on your own, must include two or more algorithms, and must integrate mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program. (Must not exceed 200 words)

```
directory = os.path.dirname(os.path.abspath(__file__))
picture = os.path.join(directory, 'Scarlett2.jpg')
img = plt.imread(picture)

height = len(img)
width = len(img[0])
for r in range(height):
for c in range(width):
if sum(img[r][c])<450: #3*255=765
img[r][c]=[253,100,0]

fig, ax = plt.subplots(1,1)
```

The first algorithm finds the filepath of the where the file is stored, and opens an image that is selected in that file path. The image has to be in that file path, then the second algorithm will give the image selected that same filepath. The third algorithm the finds the sum of the RGB values in the image. If that sum is less than 450, then those pixels will be turned into an orange color with the values of (345, 100, 0) accordingly. This sets a fine orangeness to whatever image is chosen. Finally, it is opened with the new settings applied, with a row and column axis and set. I have also been able to set different color filters on the same image in a row.

2d. Capture and paste a program code segment that contains an abstraction you developed individually on your own (marked with a rectangle in section 3 below). This abstraction must integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program. (Must not exceed 200 words)

```
height = len(img)
width = len(img[0])
for r in range(height):
for c in range(width):
if sum(img[r][c])<450: #3*255=765
img[r][c]=[253,100,0]
```

A segment of code that I developed was how to change the RGB values in order to affect only certain parts of an image. I had to look up multiple examples of image manipulations in order to find something that I could use and make the code work.

3. Program Code Capture and paste your entire program code in this section. › Mark with an oval the segment of program code that implements the algorithm you created for your program that integrates other algorithms and integrates mathematical and/or logical concepts. › Mark with a rectangle the segment of program code that represents an abstraction you developed. › Include comments or acknowledgments for program code that has been written by someone else.

```
# -*- coding: utf-8 -*-
import matplotlib.pyplot as plt
import os.path

directory = os.path.dirname(os.path.abspath(__file__))
picture = os.path.join(directory, 'Scarlett2.jpg')
img = plt.imread(picture)

height = len(img)
width = len(img[0])
for r in range(height):
for c in range(width):
if sum(img[r][c])<450: #3*255=765
img[r][c]=[253,100,0]
```

```
fig, ax = plt.subplots(1,1)
ax.imshow(img, interpolation='bicubic')
fig.show()
fig.canvas.draw()
```

A lot of the code that I used was also used for the PLTW course, and I used that as an enormous resource in the creation of my code.