* + Give a detailed explanation of how your algorithm works.

My algorithm for embedding the image with a message works as such:

- loops through each x-direction pixel

-loops through each y-direction pixel

-if the pixel number matches the next prime, alter the last 3 digits of the color’s RGB value to hide the next character of the message’s integer representation

My algorithm for retrieving the message from the image works as such:

- loops through each x-direction pixel

-loops through each y-direction pixel

-if the pixel number matches the next prime, take the last three digits of the pixel’s color’s RGB value and convert it into it’s character representation, then add it to the message string

* + Outline any help you received from classmates or other students

The only “help” I received was in discussing how to get my class to recognize the picture class form the Neumont include, an issue on which we both agreed that we could just copy and paste it into the project.

* + Mathematically analyze the performance of your algorithm using big-Oh notation. Show that your algorithm performs at or better than O(*width\*height*).

The performance of my algorithm is in the order of O(n\*n). It performs at the order of O(width\*height) as it has two loops, one spanning width and one height of the image, same as in the case of O(width\*height).

* + Empirically analyze the performance of your algorithm. Outline your approach (what made you decide these experiments were good? Etc.) and publish a table of your results, e.g.