**Note**: Site was closed so I could not reference the challenges. These write ups are all from memory.

**Cipher 1:**

This challenge referenced Caesar which pointed me to a Caesar cipher. I went to: <https://gchq.github.io/CyberChef/>

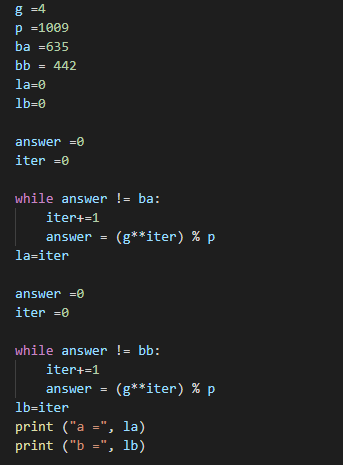
And threw the encrypted text into the ROT13 decrypter. I cycled through all 26 letters and saw the beginnings of some words but not the whole thing. Next I tried ROT 47 and cycled through. Initially I found one that kind a worked but not completely. I kept cycling through until I got the flag.

**Cipher 2:**

Simple Diffie-helmen cipher challenge. I watched this video to review:

<https://www.youtube.com/watch?v=e27HzeAVQbQ>

Wrote the following code



**Programing 2:**

Write up submitted to get the flag

**Forensics 1:**

For this challenge we were given a disk image to look through. Initially I thought this was a picture stenography challenge. I noticed the cake.jpg file and ran it through a couple different online stenography tools and got no where. I remembered the challenged mentioned something about developer sites so I started looking for developer files. In one of the folders I found a git project and a line that had flag = . This line ended in == which told me the file was base 64 encoded. So I through it into cyberchef and got the flag.

**Forensics 2:**

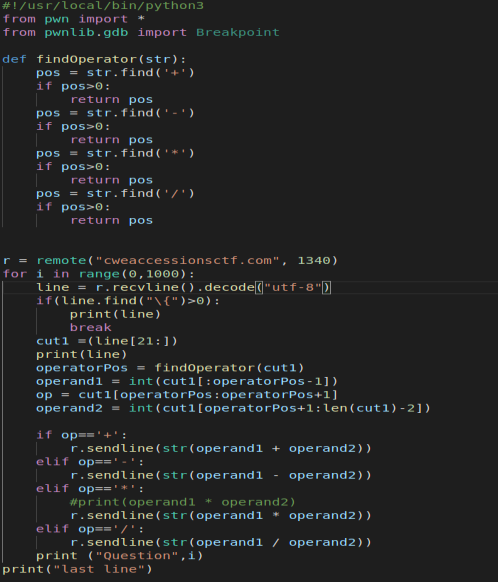
For this challenge we were given a pcap file and an ssl key. We had to configure our system and Wireshark to use the key to decrypt the encrypted traffic in the pcap file. I was familiar with Wireshark but had never used the ssl keylog file before. I watched this video:

<https://www.youtube.com/watch?v=5qecyZHL-GU>

Once decrypted I looked through the http streams and found the flag in an http2 post stream.

**Misc 1:**

For this challenge we had to automate the completion of math problems. To accomplish I wrote the following code:



**Misc 2:**

For this challenge we were given a string of characters. The title of the challenge pointed me to the least significant bit of every character. So, I copied the string into the binary converter in cyber chef. Copied that output to excel and did a data to column transformation on the last bit to isolate it. I then took the column that contained the last bit and threw it into notepad++. I did a quick replace all newline to reassemble the flag. Finally, I through the binary back into cyber chef to convert it to ascii.

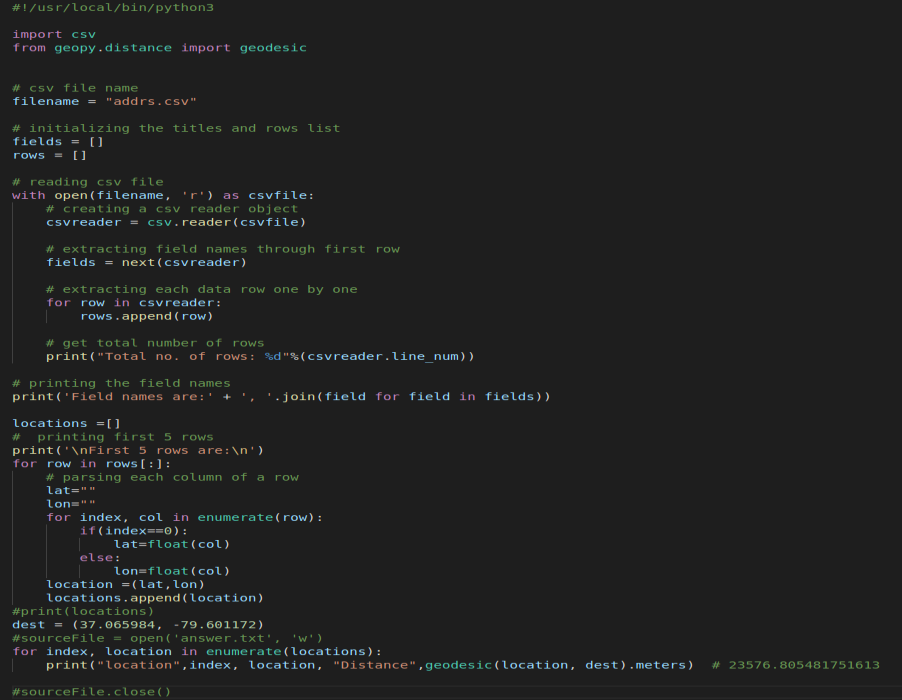
I’m sure I could have typed something up to automate this process, but it was quicker for me to do the manual manipulation. So, for time purposes I went that route.

**Misc 3:**

For this challenge we were given the hint to look at an address to coordinate API to find the closest address to a point. In order to find this answer, I knew I would have to use the distance formula and change the coordinates from degrees to radians. Rather than struggle with a new API I went to this site:

<https://www.geocod.io/upload/>

Here I formatted and uploaded the addresses and got back all the corresponding coordinates in degrees. I again manually formatted the csv file to make it pretty for the python csv module to import how I wanted it. From there I used geopy.distance in the geodesic module to calculate the distance. Below is the code I created.



**Pwn 1**

This challenge was a simple buffer overflow challenge. It was very similar to the ROP Emporium challenge 1 ret2win. I created the following code to get the flag

