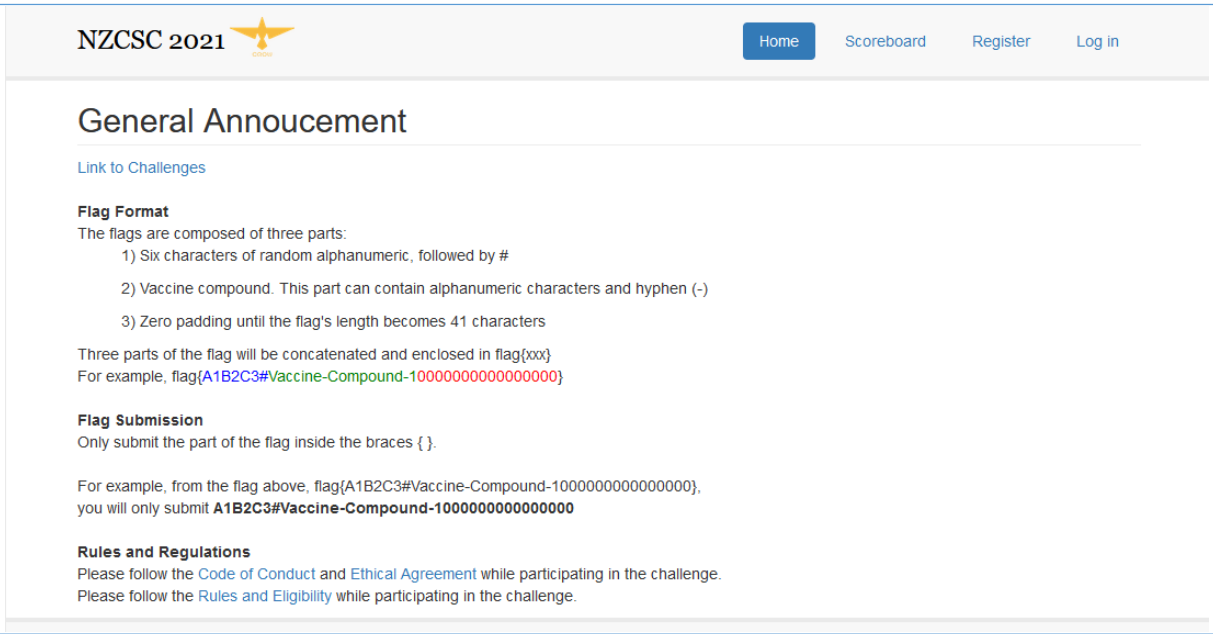
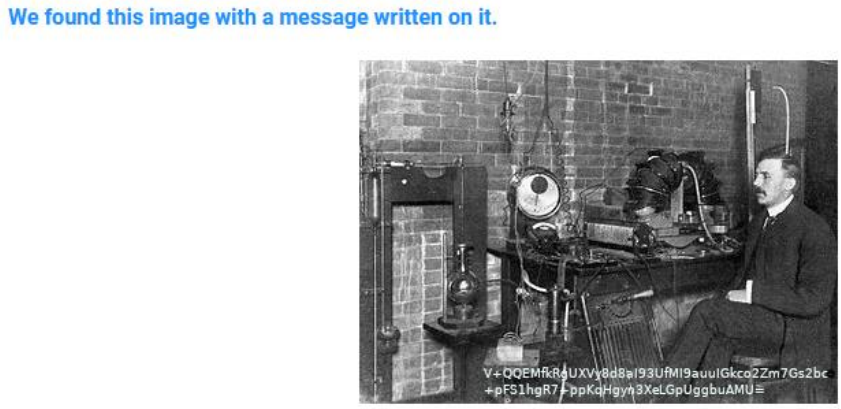


Challenge 1:



Notes:

Key components of the image are the image itself and the writing on the image which looks potentially like an encryption key.

1. To see if anything else is happening inside the image exiftool was used to analyse the image.

2. The image contained a link to its source (a Wikipedia page http://commons.wikimedia.org/wiki/File:Ernest_Rutherford_1905.jpg) so I downloaded that original image to compare in exiftool. Not much stood out.

```
Device Model Desc               : sRGB
Comment                         : File source: http://commons.wikimedia.org/wiki/File:Ernes
Image Width                     : 350
Image Height                    : 251
Encoding Process                : Progressive DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:4:4 (1 1)
Image Size                      : 350x251
Megapixels                      : 0.088
Thumbnail Image                 : (Binary data 13594 bytes, use -b option to extract)
```

3. At this point I went to the Wikipedia article and read up on Ernest Rutherford. I found the section where the image was taken from, a section about his discovery of the Rutherford model of the atom (which had a notable focus on the discovery of an interior).

4. Next I extracted the binary thumbnail (*exiftool.exe c.jpg –b*). It looked like an ordinary file. Extracting it as an image proved similar (*exiftool.exe -a -b -W NEW/%f_%t%-c.%s -preview:all c.jpg*).

5. I then inspected the binary of the image. And opened the image in gimp, where it had been edited. I also opened the image in a hex editor to see if there was anything I was missing.

6. It was time to take a look at the text written on the image.

```
V+QQEMfkRgUXVy8d8aI93UfMI9auuIGkco2Zm7Gs2vc
+pFS1hgR7+ppKoHgyn3XeLGpUggbuAMU=
```

7. My first guess was that this could be a key. But because “a message on it” is listed there is a reasonable probability it is an encoded message. The first line was 44 characters long, the second line was 34. Another plausible option is that either line could be a hash code. They aren’t quite formatted correctly though.

   Since + and = are the only symbols we likely are either using base64 or they hold some significance as separators and delimiters.

8. At this point I decided to take another approach and ran xortool, a powerful cypher decryption tool on the two lines given.

```
┌──(resti㉿ MSI)-[~/Projects]
└─$ xortool -b -p "flag" XOR.bin
The most probable key lengths:
 1:  17.4%
 4:  19.7%
 6:  21.7%
10:  14.9%
15:   7.0%
18:   6.1%
20:   3.7%
22:   3.4%
24:   3.2%
26:   2.9%
Key-length can be 3*n
Key-length can be 4*n
6912 possible key(s) of length 6:
Vk3g2M
Vk3U2M
Vk3o2M
V93g2M
V93U2M
```

9. I further consider the option of the words being base64 (any smaller base system ruled out because of the number of characters). I run through some conversions. Base64 to plaintext/hex appears mostly garbage. An Xor bruteforce attack doesn't return any results.

Hex Dump

```
          00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17
00000000  57 e4 10 10 c7 e4 46 05 17 57 2f 1d f1 a2 3d dd 47 cc 23 d6 ae b8 81 a4  W.....F..W/...=.G.#.....
00000018  72 8d 99 9b b1 ac da f7 3e a4 54 b5 86 04 7b fa 9a 4a a0 78 32 9f 75 de  r.......>.T...{..J.x2.u.
00000030  2c 6a 54 82 06 ee 00 c5                                                  ,jT.....
```

10. A clue given for the challenge lists the encoded text requires a key of length 12. This makes me consider XOR further.

Hint! The encoded text requires a key of length 12.

Incomplete…