Challenge Lab 3
UWYO COSC 2030

1 Challenge Lab 3

1.1 Steganography

The practice of hiding a secret message in something that is not secret

1.2 The Lab

Given the supplied encoded_apollo.jpg your task is to decode the message written inside.
This lab can be done in either C++ or Python. Personally I think it is more easily accomplished in Python, but ultimately whatever you are most comfortable with is the better choice. I have written solutions in both languages, so it is do-able in both.

1.3 The Layout

The string is encoded on the least significant (the right most) bit of each byte in a continuous block of bytes in the picture. Each character is made up of 8 bits, so each character takes a block of 64 bits, 8 bytes, to produce. Only lowercase characters, spaces, new line characters, and exclamation points are encoded in the picture. The string is prefaced by "!!!" and ended by the same deliminator. This is so you know when to start printing the string to the terminal, so for example an encoded string would look like

```
!!! this is an example string
it goes on multiple lines !!!
```

Where each character, including the spaces and new line character, are encoded in the picture over 8 bytes each.

1.4 Output

Your output should not include the exclamation marks. There is only one encoded string denoted by the set of the exclamation marks on each side. This means once you reach the second set of exclamation marks you can stop going through the bytes. Your output should be EXACTLY what is encoded, in that you should only output a space or new line character if the encoded character is one of those two options. Otherwise, output the alphabetic character in lowercase to the terminal. This means do not output the exclamation marks or anything else.

1.5 Things to Note

1. You are only pulling the least significant bit off the needed bytes. This is the right most bit
2. Bytes are encoded with Little-Endian
3. Characters are encoded with ASCII
4. The encoding starts arbitrarily somewhere in the picture, I don't even know where, but it is an offset of 8
5. After that you can assume that blocks of 8 bytes can be pulled to get the characters of the string
6. The string within the "!!!" markers does not contain any punctuation
7. It will be obvious when you have found the string
8. As you read in 8-byte blocks, you can assume the first byte contains the leftmost bit and the final one contains the rightmost bit of the ASCII byte

1.6 Suggestions

1. You should read in blocks of 8 bytes at time. The encoding, while randomly placed, is offset by a multiple of 8
2. The little endian-ness only matters if you use an encode function that needs to know if it is "big" or "little", you do not need to deal with 4-bytes of data at any point
3. Verify the md5sum of the image file:
   (a) Place your picture file in the linux directory you will be coding in
   (b) On the terminal run ther command

   ```
   md5sum encoded_apollo.jpg
   ```

   (c) your output should match below, if it doesn't message Danny to get the picture again

   ```
   cfe1d97c28dcadd2b9fbee60afe49ae3  encoded_apollo.jpg
   ```

4. Make sure you are paying attention to the typing of things, especially if using python. Python will let you throw things around without knowing the type. If you ever need to know or want to check you can add:

   ```
   print(type(whatever-you-are-checking))
   ```

5. When you read in the bytes remember only the least significant bit is important (for this), the other 7 can be ignored
6. You aren't expected to know how to do this, but you are expected to be able to identify the problem and research it
7. Yes, this lab is hard.

1.7 Submission

1. Your file MUST follow this naming stand:

   ```
   YourLastNameChal3.cpp OR YourLastNameChal3.py
   ```

2. Your code should output only the encoded string. If you print out the exclamation marks I will take points. You have been warned.
3. DO NOT add any additional spaces or newline characters. Only print those that are encoded
4. If a space or newline character is encoded you have to print them, I do not want one long string without spaces or newlines
5. Upload your file to the jump cloud folder. Submissions WILL NOT be accepted after 5:30pm for the in-person session and after 7:50pm for the digital
6. When done please upload your file to the nextcloud link. Please only do so once.