# Hash Problem

Whilst on the Cyberman spaceship, Graham comes across a terminal. A sticky note near the terminal says, “The credentials for the next account are in file.” Graham lists the files in the terminal and sees 100 files, file0.txt through file99.txt. He looks back at the note to see which file he needs, but the sticky note was torn off just after “file”.

Help Graham find the file he needs. All but one of the files are identical; the different file is the one you want. A simple loop that takes the hash of each file is one way to do it. It does not matter which form of hash (MD5, SHA-1, etc.) you use, as we do not expect malicious files that create hash collisions. (There are other ways to solve this if you do not like hashes.)

The files are in [**hashfiles.zip**](https://learn.vccs.edu/courses/314236/files/92777927/preview).  There are 100 files, so put the zip file in its own subdirectory before you unzip.  It could make a mess.  Just sayin’.

**What is the content of the file that contains the credentials?**

Easy way to decode hints: <https://gchq.github.io/CyberChef/#recipe=From_Base64('A-Za-z0-9%2B/%3D',true,false)>

Terminal Hint 1: Both Linux (md5sum, sha256sum, etc.) and Windows (Get-Filehash) will allow you to use a wild card (\*) in place of the filename to take the hash of every file in a directory. Once you have all the hashes printed out, it is easy to see which file is different.

Qm90aCBMaW51eCAobWQ1c3VtLCBzaGEyNTZzdW0sIGV0Yy4pIGFuZCBXaW5kb3dzIChHZXQtRmlsZWhhc2gpIHdpbGwgYWxsb3cgeW91IHRvIHVzZSBhIHdpbGQgY2FyZCAoKikgaW4gcGxhY2Ugb2YgdGhlIGZpbGVuYW1lIHRvIHRha2UgdGhlIGhhc2ggb2YgZXZlcnkgZmlsZSBpbiBhIGRpcmVjdG9yeS4gIE9uY2UgeW91IGhhdmUgYWxsIHRoZSBoYXNoZXMgcHJpbnRlZCBvdXQsIGl0IGlzIGVhc3kgdG8gc2VlIHdoaWNoIGZpbGUgaXMgZGlmZmVyZW50Lg==

Python Hint 1: If you want to do hashes in Python, it is easy. If you have PyCryptodome installed, you can use this:  
from Crypto.Hash import MD5  
MD5.new(b'this is what I want to hash').hexdigest()

If you don’t have PyCryptodome installed, this will work:  
import hashlib  
hashlib.md5(b'this is what I want to hash').hexdigest()  
Note: the input must be type bytes, b';dlfja;s' or variable.encode()



Python Hint 2

import hashlib

with open('file0.txt') as fh:

content = fh.read().encode()

myhash = hashlib.md5(content).hexdigest()

print(myhash)  
  
aW1wb3J0IGhhc2hsaWIKCndpdGggb3BlbignZmlsZTAudHh0JykgYXMgZmg6CiAgICBjb250ZW50ID0gZmgucmVhZCgpLmVuY29kZSgpCm15aGFzaCA9IGhhc2hsaWIubWQ1KGNvbnRlbnQpLmhleGRpZ2VzdCgpCnByaW50KG15aGFzaCk=

Another way Hint:  
Find a word in the first file that you guess might not be in the file that’s different  
Use grep to find a file that doesn’t have the word you chose  
There is a flag in grep for ‘not match’  
RmluZCBhIHdvcmQgaW4gdGhlIGZpcnN0IGZpbGUgdGhhdCB5b3UgZ3Vlc3MgbWlnaHQgbm90IGJlIGluIHRoZSBmaWxlIHRoYXTigJlzIGRpZmZlcmVudApVc2UgZ3JlcCB0byBmaW5kIGEgZmlsZSB0aGF0IGRvZXNu4oCZdCBoYXZlIHRoZSB3b3JkIHlvdSBjaG9zZQpUaGVyZSBpcyBhIGZsYWcgaW4gZ3JlcCBmb3Igbm90IG1hdGNoClVzZSB0aGUgLXIgZmxhZyBmb3IgcmVjdXJzaXZl