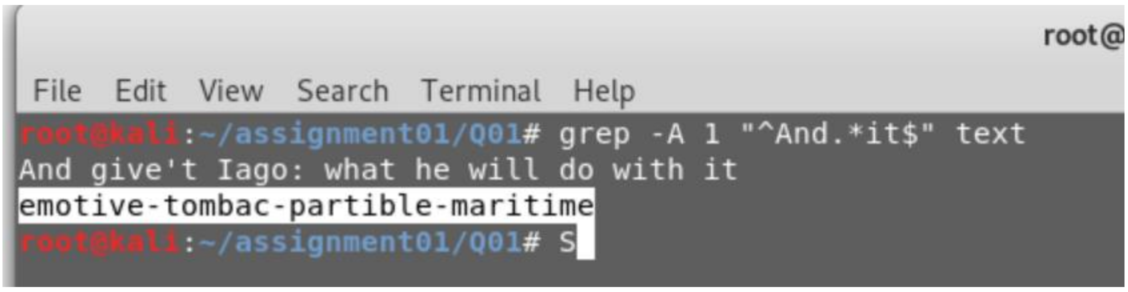
Cybersecurity Fundamentals (CS3308/7308) Assignment 1 Example Answers

Question 1

There is a secret passphrase embedded in the file called "text" in the folder Q01. The secret can be found following the line that begins with the word "**And**" and ends with "**it**". Use grep with regular expression to locate this line and the line following it. What is the passphrase? Hint: look up the use of the -A option with grep.

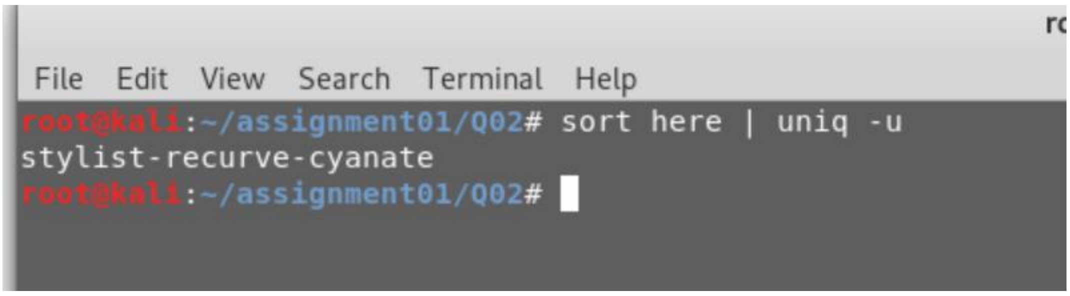


```
root@kali:~/assignment01/Q01# grep -A 1 "^And.*it$" text
And give't Iago: what he will do with it
emotive-tombac-partible-maritime
root@kali:~/assignment01/Q01# S
```

- The "-A *n*" option prints n lines after the matched line(s). Similarly, -B gives you lines before, and –C gives you lines before and after

Question 2

There is a file called "here" that contains passphrases. Find the passphrase that occur exactly ONCE. Remember to sort before uniq.

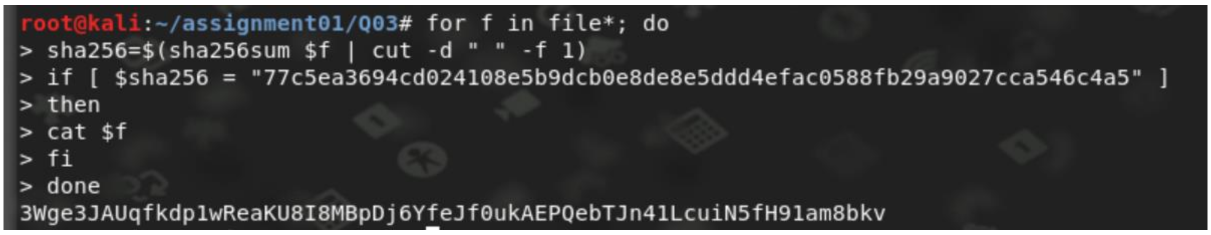


```
root@kali:~/assignment01/Q02# sort here | uniq -u
stylist-recurve-cyanate
root@kali:~/assignment01/Q02#
```

- Uniq command assumes the input is already sorted, so you just need to pipe the output of sort (ascending or descending does not matter) to uniq.
- Uniq command gives you the number of occurrences, and the "-u" options just prints the singletons

Question 3

There are lots of files here. What is the **content** of the file whose SHA256 sum is 77c5ea3694cd024108e5b9dcb0e8de8e5ddd4efac0588fb29a9027cca546c4a5? Write a script to find out.



```
root@kali:~/assignment01/Q03# for f in file*; do
> sha256=$(sha256sum $f | cut -d " " -f 1)
> if [ $sha256 = "77c5ea3694cd024108e5b9dcb0e8de8e5ddd4efac0588fb29a9027cca546c4a5" ]
> then
> cat $f
> fi
> done
3Wge3JAUqfkdp1wReaKU8I8MBpDj6YfeJf0ukAEPQebTJn41LcuiN5fH91am8bkv
```

- The matching file was file00072
- A much elegant way (thanks Mr Ian Crossing) was to do this:

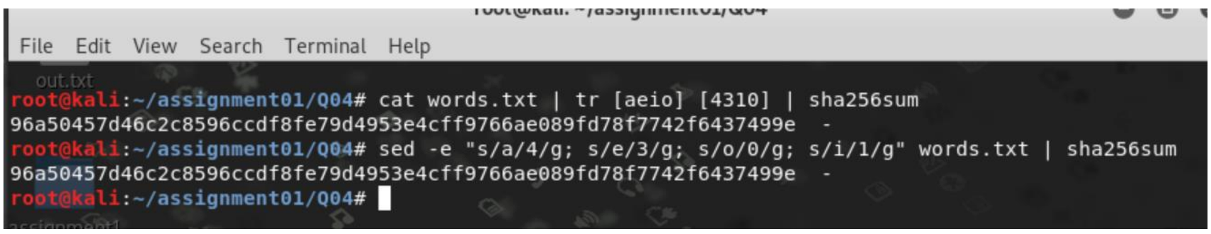
```
sha256sum * | grep 77c5ea3694cd024108e5b9dcb0e8de8e5ddd4efac0588fb29a9027cca546c4a5
```

You could then create a one-liner like this to reveal the content of the file

```
cat $(sha256sum * | grep \
77c5ea3694cd024108e5b9dcb0e8de8e5ddd4efac0588fb29a9027cca546c4a5 \
| awk -F ' ' '{print $2}')
```

Question4

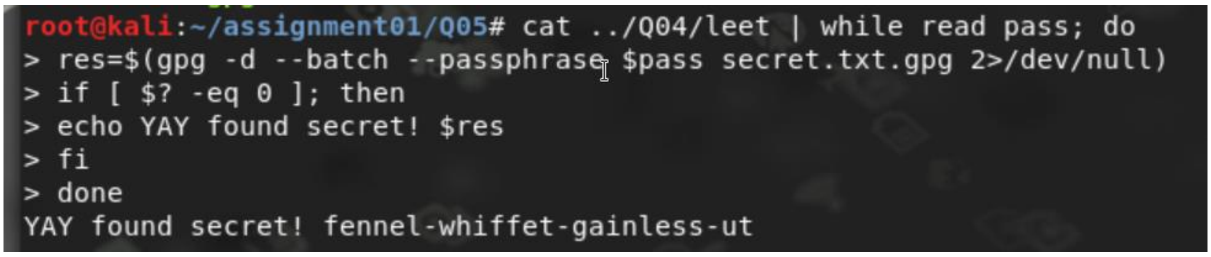
Use the tr (transform) command to generate a list of passwords from the source file words.txt. Use "l33t" conversion so that a=>4, e=>3, i=>1, and o=>0. For example "hello world" becomes "h3ll0 w0rld". Calculate the SHA256 hash of the generated file and provide that as the answer. You could also use sed instead of tr.



```
root@kali:~/assignment01/Q04# cat words.txt | tr [aeio] [4310] | sha256sum
96a50457d46c2c8596ccdf8fe79d4953e4cff9766ae089fd78f7742f6437499e  -
root@kali:~/assignment01/Q04# sed -e "s/a/4/g; s/e/3/g; s/o/0/g; s/i/1/g" words.txt | sha256sum
96a50457d46c2c8596ccdf8fe79d4953e4cff9766ae089fd78f7742f6437499e  -
root@kali:~/assignment01/Q04# 
```

Question 5

There is a file encrypted using gpg (Gnu Privacy Protection), using the command gpg -c --passphrase <pass>. Unfortunately we have forgotten the password. Try the "l33t" converted password list from Question 4 to brute-force the encrypted file. What is the secret? Here is a skeleton for a bash script. Remember you can ignore error messages by doing 2>/dev/null. This could take a few minutes...

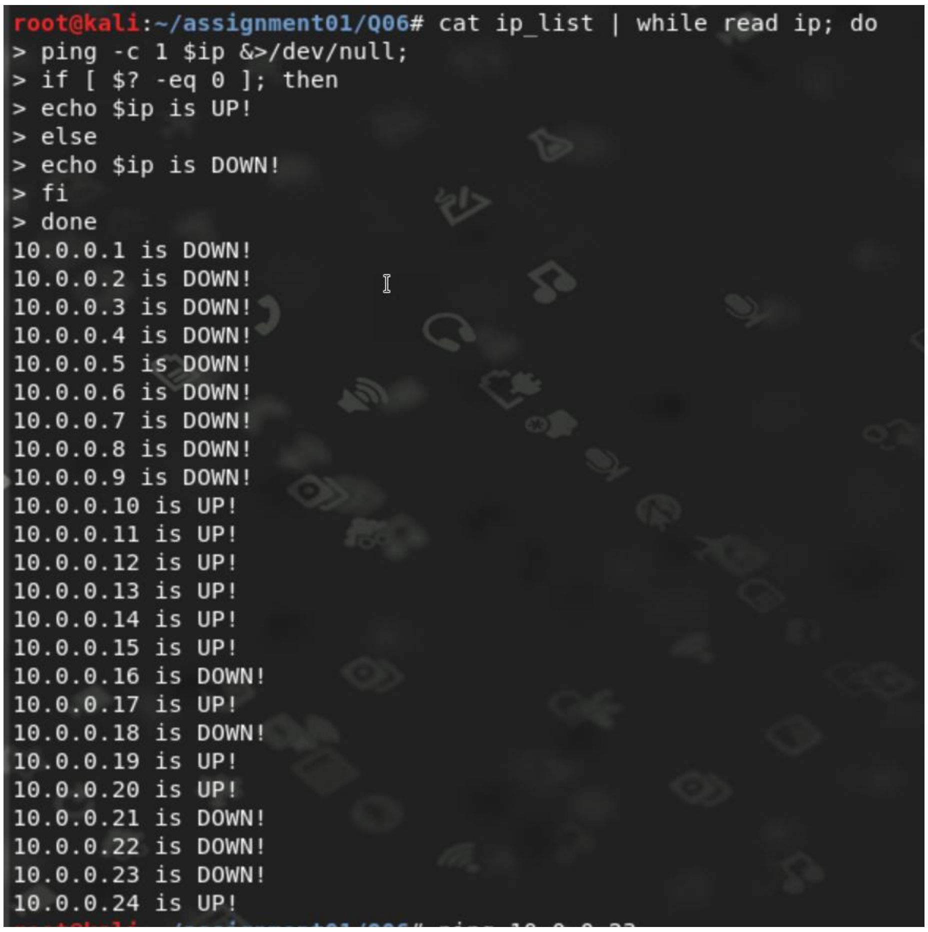


```
root@kali:~/assignment01/Q05# cat ../Q04/leet | while read pass; do
> res=$(gpg -d --batch --passphrase $pass secret.txt.gpg 2>/dev/null)
> if [ $? -eq 0 ]; then
> echo YAY found secret! $res
> fi
> done
YAY found secret! fennel-whiffet-gainless-ut
```

- The only tricky bit was to use the "--batch --passphrase" modifier to prevent the interactive passphrase entry.

Question 6

You can use ping to find if a host is up (some hosts don't respond to ping, but that's OK for the purpose of this exercise). Write a script that pings each host in the **ip_list** file in Q06 folder, and prints "<ip> is up!" or "<ip> is down!" for each IP in the file based on the response code (in bash, the response code of the previous command is "$?". For ping 0 means the host responded, 1 means it did not respond, and 2 is some error. Use the "-c 1" modifier to ping just once. Also, it is useful to send all output to /dev/null.

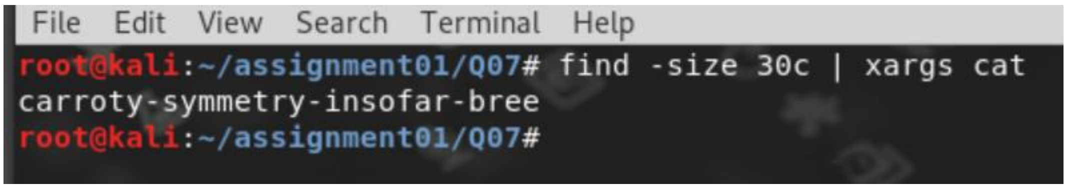


```
root@kali:~/assignment01/Q06# cat ip_list | while read ip; do
> ping -c 1 $ip &>/dev/null;
> if [ $? -eq 0 ]; then
> echo $ip is UP!
> else
> echo $ip is DOWN!
> fi
> done
10.0.0.1 is DOWN!
10.0.0.2 is DOWN!
10.0.0.3 is DOWN!
10.0.0.4 is DOWN!
10.0.0.5 is DOWN!
10.0.0.6 is DOWN!
10.0.0.7 is DOWN!
10.0.0.8 is DOWN!
10.0.0.9 is DOWN!
10.0.0.10 is UP!
10.0.0.11 is UP!
10.0.0.12 is UP!
10.0.0.13 is UP!
10.0.0.14 is UP!
10.0.0.15 is UP!
10.0.0.16 is DOWN!
10.0.0.17 is UP!
10.0.0.18 is DOWN!
10.0.0.19 is UP!
10.0.0.20 is UP!
10.0.0.21 is DOWN!
10.0.0.22 is DOWN!
10.0.0.23 is DOWN!
10.0.0.24 is UP!
```

- You can basically use the same script structure as Question 5

Question 7

There are lots of sub-directories and files under Q07. Find the file containing the secret. The file has size of exactly 30 bytes long. Hint: pipe find into "| xargs cat" to save time.

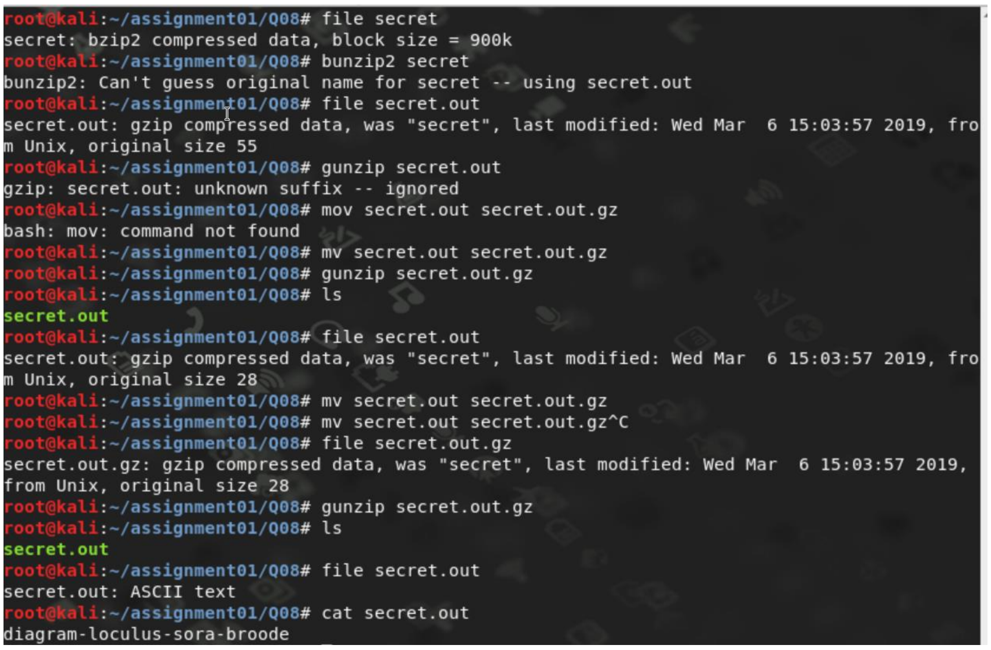


```
root@kali:~/assignment01/Q07# find -size 30c | xargs cat
carroty-symmetry-insofar-bree
root@kali:~/assignment01/Q07#
```

- Pretty easy with "find" command and the "-size" modifier. The "c" in "30c" indicates 30 bytes, remembering that char is an 8-bit variable.

Question 8

The secret is in Q08 folder. Hint: use the file command to determine the type of file

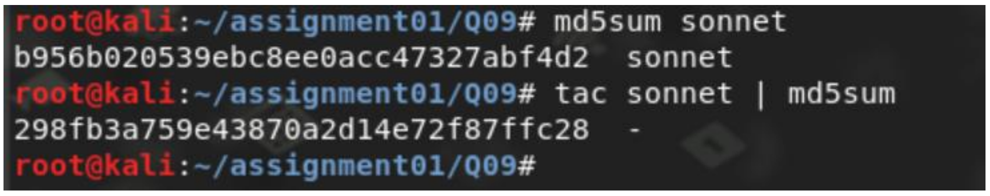


```
root@kali:~/assignment01/Q08# file secret
secret: bzip2 compressed data, block size = 900k
root@kali:~/assignment01/Q08# bunzip2 secret
bunzip2: Can't guess original name for secret -- using secret.out
root@kali:~/assignment01/Q08# file secret.out
secret.out: gzip compressed data, was "secret", last modified: Wed Mar  6 15:03:57 2019, fro
m Unix, original size 55
root@kali:~/assignment01/Q08# gunzip secret.out
gzip: secret.out: unknown suffix -- ignored
root@kali:~/assignment01/Q08# mov secret.out secret.out.gz
bash: mov: command not found
root@kali:~/assignment01/Q08# mv secret.out secret.out.gz
root@kali:~/assignment01/Q08# gunzip secret.out.gz
root@kali:~/assignment01/Q08# ls
secret.out
root@kali:~/assignment01/Q08# file secret.out
secret.out: gzip compressed data, was "secret", last modified: Wed Mar  6 15:03:57 2019, fro
m Unix, original size 28
root@kali:~/assignment01/Q08# mv secret.out secret.out.gz
root@kali:~/assignment01/Q08# mv secret.out secret.out.gz^C
root@kali:~/assignment01/Q08# file secret.out.gz
secret.out.gz: gzip compressed data, was "secret", last modified: Wed Mar  6 15:03:57 2019,
from Unix, original size 28
root@kali:~/assignment01/Q08# gunzip secret.out.gz
root@kali:~/assignment01/Q08# ls
secret.out
root@kali:~/assignment01/Q08# file secret.out
secret.out: ASCII text
root@kali:~/assignment01/Q08# cat secret.out
diagram-loculus-sora-broode
```

- Just use the file command repeatedly to find the file type, then use the decompression tool suited for the file format

Question 9

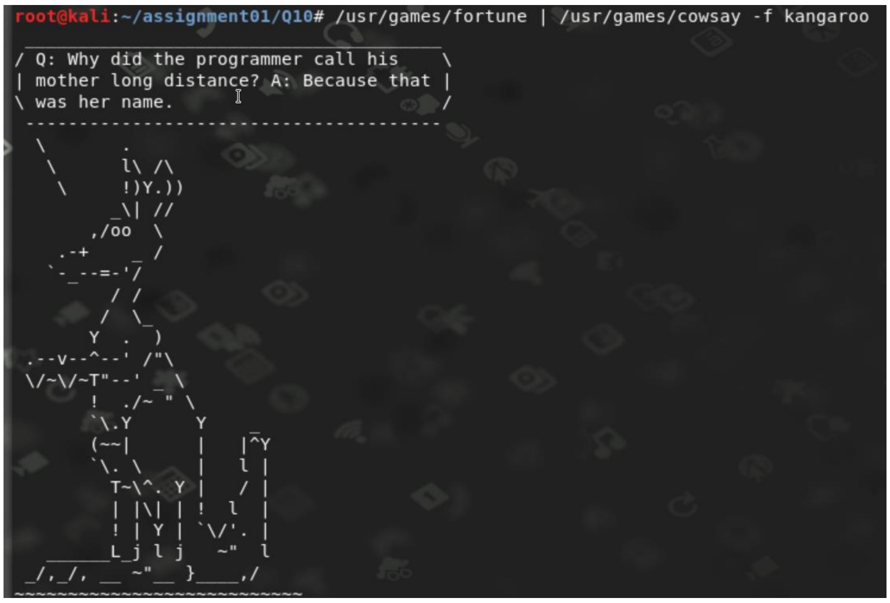
There is a poem in Q09 folder. Calculate the SHA256 hash of the poem as is, and also when the lines are reversed (i.e, provide 2 hashes).



```
root@kali:~/assignment01/Q09# md5sum sonnet
b956b020539ebc8ee0acc47327abf4d2  sonnet
root@kali:~/assignment01/Q09# tac sonnet | md5sum
298fb3a759e43870a2d14e72f87ffc28  -
root@kali:~/assignment01/Q09#
```

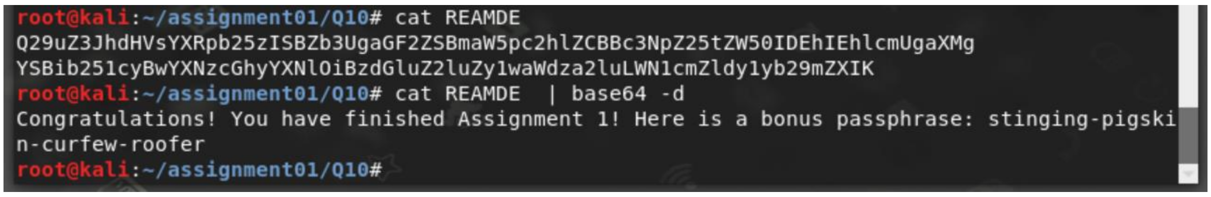
Question 10

Run this command and paste the result as evidence that you have done the workshop. See /usr/share/cowsay/cows for the list of characters.



```
root@kali:~/assignment01/Q10# /usr/games/fortune | /usr/games/cowsay -f kangaroo
 ________________________________________
/ Q: Why did the programmer call his     \
| mother long distance? A: Because that  |
\ was her name.                          /
 ----------------------------------------
  \       .
   \      l\ /\
    \     !)Y.))
         _\| //
       ,/oo  \
    .-+     _ /
   `-_--=-'/
         / /
        /  \_
       Y  .  )
 .--v--^--' /"\
 \/~\/~T"--' _ \
       !  ./~ " \
       `\.Y      Y    _
       (~~|      |   |^Y
       `\. \     |   l |
         T~\^. Y |   / |
         | |\| | !  l  |
         ! | Y | `\/'. |
   ______L_j l j   ~"  l
 _/,_/, __ ~"__ }____,/
~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Hidden bonus question



```
root@kali:~/assignment01/Q10# cat REAMDE
Q29uZ3JhdHVsYXRpb25zISBZb3UgaGF2ZSBmaW5pc2hlZCBBc3NpZ25tZW50IDEhIEhlcmUgaXMg
YSBib251cyBwYXNzcGhyYXNlOiBzdGluZ2luZy1waWdza2luLWN1cmZldy1yb29mZXIK
root@kali:~/assignment01/Q10# cat REAMDE  | base64 -d
Congratulations! You have finished Assignment 1! Here is a bonus passphrase: stinging-pigski
n-curfew-roofer
root@kali:~/assignment01/Q10#
```