Assignment 1

1)

For this problem I stored the "text" in a file call the passphrase is the line just after the line that begins with the word "And" and end with "it". So using the grep -A it will output the line(s) trailing the when matches the criteria set according to the question.

```
a1715329@kali:~/Desktop/assignment01/Q01$ grep -A 1 ^And.*it$ text
And give't Iago: what he will do with it
emotive-tombac-partible-maritime
```

Answer: emotive-tombac-partible-maritime

2)
In this problem there is file called "here" that has a passphrase. Since there is lot of passphrases that are not in order and repeated we use the sort(to sort the passphrases), uniq(filter out repeating lines in a file) command and pipes to get the required passphrases that occur exactly once.

Answer:

```
a1715329@kali:~/Desktop/assignment01/Q02$ cat here|sort|uniq
advowson-stag-bushel
appetite-twin-fid
benedict-parlance-algid
bugloss-holism-barbecue
caboose-batting-frolic
catering-amrita-piscina
cease-crater-quietly
collage-markka-ally
cracking-sip-ye
enrage-operetta-foofaraw
eolithic-trestle-aneroid
eucharis-soluble-snicker
external-paravane-bizarre
finespun-archive-impede
front-tailgate-cilice
fryer-lambdoid-parquet
gesture-saran-elf
hayseed-liberty-dredge
jeepers-tried-twill
jetport-eugenol-jemmy
kismet-camphene-sluggard
leave-joyous-snath
lineman-calabash-voyage
mode-stemson-cracker
obloquy-micelle-perique
oversold-sward-cesta
points-mastic-delta
saturate-smithery-rood
senses-blew-totter
shamble-textuary-fiche
stern-lip-fake
stricken-raze-overcrop
stylist-recurve-cyanate
sulphur-weave-way
supine-illinium-mandrill
tanked-nimbus-pinch
thicket-seacoast-sannyasi
threnode-dispend-cuboid
transfix-grummet-terry
vexed-poultice-comical
woofer-jagged-mir
```

3)

For this problem we use a for loop to determine which file has the same sha256sum. The file that matches display OK and rest will be failed and will show a warning.

```
#!/bin/bash
for j in $( ls ); do
        echo "$j"
echo "77c5ea3694cd024108e5b9dcb0e8de8e5ddd4efac0588fb29a9027cca546c4a5 $j" | sha256sum --check
done
```

```
a1715329@kali:~/Desktop/assignment01/Q03$ ./q3.sh
file00001
file00001: FAILED
sha256sum: WARNING: 1 computed checksum did NOT match
file00002
file00002: FAILED
sha256sum: WARNING: 1 computed checksum did NOT match
```

```
file00071
file00071: FAILED
sha256sum: WARNING: 1 computed checksum did NOT match
file00072
file00072: OK
file00073
file00073: FAILED
sha256sum: WARNING: 1 computed checksum did NOT match
```

So we see that file0072 is the file that has the same sha256sum. Then we use cat to show the content of the file.

Answer:

```
a1715329@kali:~/Desktop/assignment01/Q03$ cat file00072
3Wge3JAUqfkdp1wReaKU8I8MBpDj6YfeJf0ukAEPQebTJn41LcuiN5fH91am8bkv
```

4)

For this problem I used the sed -i (stream editior and -i saves backups after editing).

```
a1715329@kali:~/Desktop/assignment01/Q04$ sed -i 's/o/0/g' words.txt
a1715329@kali:~/Desktop/assignment01/Q04$ sed -i 's/e/3/g' words.txt
a1715329@kali:~/Desktop/assignment01/Q04$ sed -i 's/i/1/g' words.txt
a1715329@kali:~/Desktop/assignment01/Q04$ sed -i 's/a/4/g' words.txt
a1715329@kali:~/Desktop/assignment01/Q04$ cat words.txt
```

This is the output of the word file after converting according to the "I33t" conversion.

```
m4rk
supp0s3
4cc3pt4bl3
bus1n3ss
34rspl1tt1ng
l3v3l
l4ck4d41s1c4l
qu1ll
3v4n3sc3nt
gr4t3ful
3r3ct
0b3s3
sm3ll
st33p
br4k3
pl0t
f4d3
sk4t3
r1p3
gr0uchy
sc4r3cr0w
04f1sh
m4l1c10us
v31n
p4rt
4b4ft
4tt4ch
h3lpful
3c0n0m1c
ch1v4lr0us
r0d
d3l1c10us
f41thful
d3r4ng3d
s0und
sc4r3
sw1ft
sp3ct4cul4r
s1nc3r3
d0ll
1ncr3d1bl3
```

After this I found the sha256 sum of the converted file according to the question:

Answer:

```
a1715329@kali:~/Desktop/assignment01/Q04$ sha256sum words.txt
65a2aa3aa81529a60a13236d9688619f9151a6b98fc0105491ee111e0b5b6058  words.txt
```

5)

In this we to try to get the passphrase and if gpg executes and it returns) then echo "YAY found secret!".

```bash
#!/bin/bash
# The script assumes that the passwords are stored in a file called 'word.txt'
# Note that after gpg executes you can check the result using the return code variable $?

cat words.txt | while read line; do
    res=$(gpg --passphrase $line --pinentry-mode loopback secret.txt.gpg 2>/dev/null)
    if [ 0 -eq $? ] # if the return value is 0, then no error
    then
        echo YAY found secet! $res
        exit
    fi
done
```

```
a1715329@kali:~/Desktop/assignment01/Q05$ ./q5.sh
YAY found secet!
```

The content inside secret.txt:

```
fennel-whiffet-gainless-ut
```

6)

This is the script to determine if the ip is up, down or error. We read each ip from ip_list and echo the appropriate result. Also to get the correct result we should connect hacklab vpn.

```bash
#!/bin/bash

cat ip_list | while read line
do
        IP=$line
        fping -c1 -t300 $IP 2>/dev/null 1>/dev/null
        if [ "$?" = 0 ]
        then
          echo "<ip> is up!"

        elif [ "$?" = 2 ]
        then
          echo "error"

        else
          echo "<ip> is down!"
        fi
done
```

```
a1715329@kali:~/Desktop/assignment01/Q06$ ./q6.sh
<ip> is down!
<ip> is down!
<ip> is down!
<ip> is down!
<ip> is down!
<ip> is down!
<ip> is down!
<ip> is down!
<ip> is down!
<ip> is up!
<ip> is up!
<ip> is up!
<ip> is up!
<ip> is up!
<ip> is up!
<ip> is down!
<ip> is up!
<ip> is down!
<ip> is down!
<ip> is up!
<ip> is down!
<ip> is down!
<ip> is down!
<ip> is up!
```

7)
In this question there are a lot of sub-directories and files given. The find -size 30c (c is size in bytes) will find the file of size 30 bytes that contain the secret. Then piping it to xargs cat will given the secret embeded in the file.

Answer:

```
a1715329@kali:~/Desktop/assignment01/Q07$ find -size 30c | xargs cat
carroty-symmetry-insofar-bree
```

8)

In this question I first use the file command to determine the type of the file. Then I will keep on unziping the files until the file type becomes text. Then i will use the cat command to print the secret of the file.

Answer:

```
a1715329@kali:~/Desktop/assignment01/Q08$ ls
secret
a1715329@kali:~/Desktop/assignment01/Q08$ file secret
secret: bzip2 compressed data, block size = 900k
a1715329@kali:~/Desktop/assignment01/Q08$ bzip2 -dk <secret> sub
a1715329@kali:~/Desktop/assignment01/Q08$ ls
secret  sub
a1715329@kali:~/Desktop/assignment01/Q08$ file sub
sub: gzip compressed data, was "secret", last modified: Wed Mar  6 15:03:57 2019, from Unix, original size 55
a1715329@kali:~/Desktop/assignment01/Q08$ gzip -dk <sub> subsub
a1715329@kali:~/Desktop/assignment01/Q08$ file subsub
subsub: gzip compressed data, was "secret", last modified: Wed Mar  6 15:03:57 2019, from Unix, original size 28
a1715329@kali:~/Desktop/assignment01/Q08$ gzip -dk <subsub> subsubsub
a1715329@kali:~/Desktop/assignment01/Q08$ file subsubsub
subsubsub: ASCII text
a1715329@kali:~/Desktop/assignment01/Q08$ cat subsubsub
diagram-loculus-sora-broode
```

9)

For this question first we will find sha256sum of the poem as it is by using pipe(using output of cat sonnet as the input for find sha256sum). Then we do tac sonnet that will reverse the poem and then pipe like the first step.

Answer:

```
a1715329@kali:~/Desktop/assignment01/Q09$ cat sonnet | sha256sum
cda37e5206ef17aa8d0446090e1bbaf5e4c90d3aa0e652854c5bd4cdf266e7ef
a1715329@kali:~/Desktop/assignment01/Q09$ tac sonnet | sha256sum
670c2f90170b1b0b3c67a9bfeb00e08a48d204f67cb9a37122fa0dc80d4cb8dd
```

10)

I ran the command given and the result obtained was:

Answer: