CSI4139 LAB 2 Report

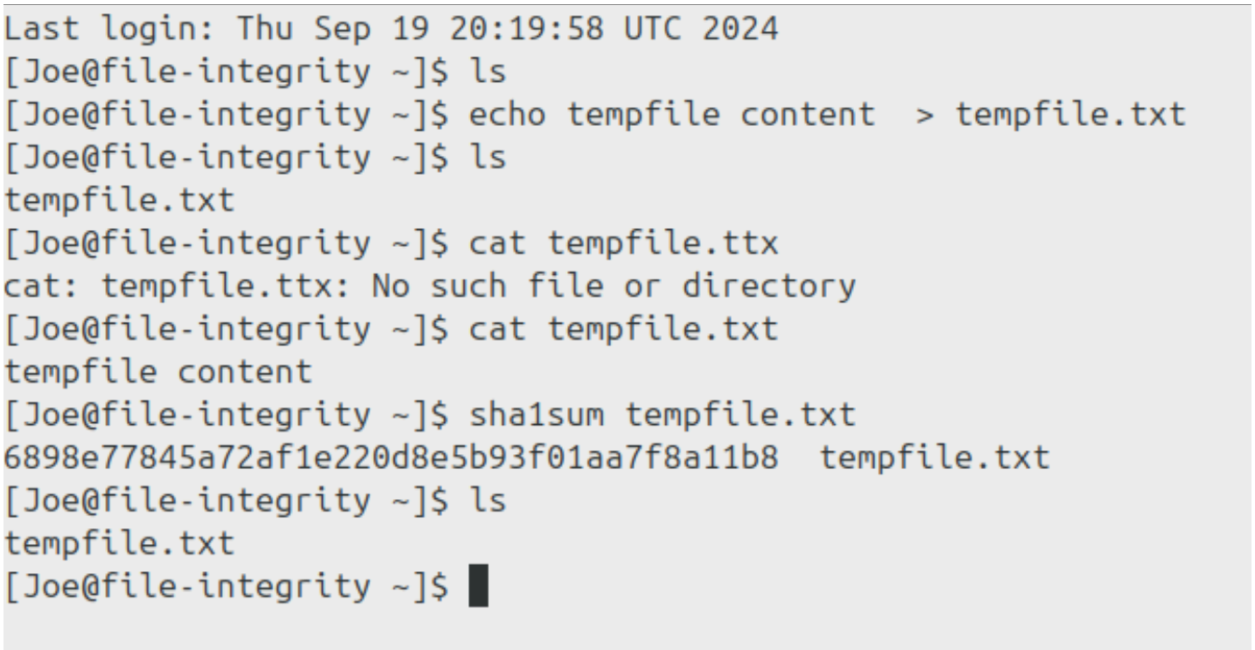
Name: Iyanu Aketepe

Student No: 300170701

Command Legend:

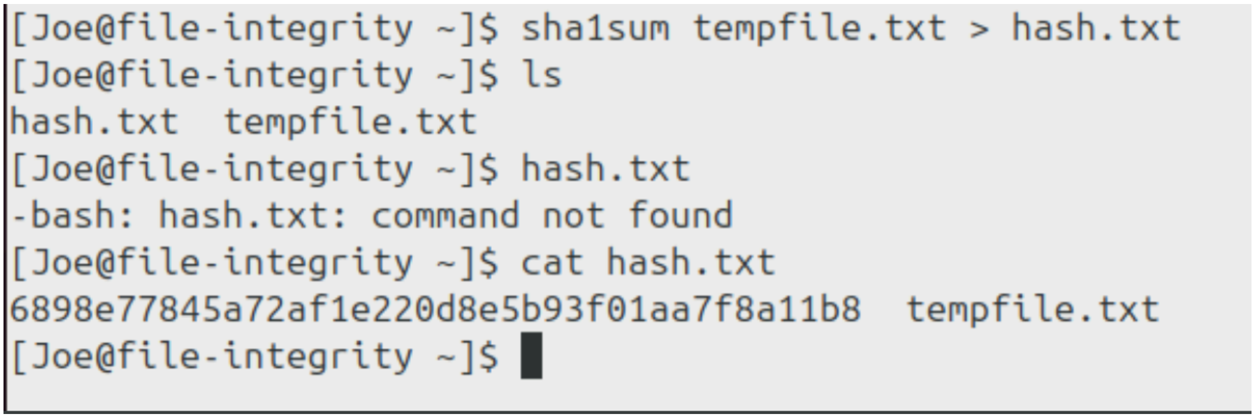
ls: Lists files and directories in the current directory.
echo: Prints text or variables to the terminal.
cat: Concatenates and displays the contents of files.
sha1sum: Generates and prints the SHA-1 hash for a file.
mkdir: Creates a new directory.
cd: Changes the current directory to the specified path.
sha1sum --check: Verifies file integrity by checking the SHA-1 hash against a provided list.
nano: Opens the Nano text editor for creating or editing files.
wc: Counts the lines, words, and characters in a file or input.

Task 1
A



```
Last login: Thu Sep 19 20:19:58 UTC 2024
[Joe@file-integrity ~]$ ls
[Joe@file-integrity ~]$ echo tempfile content  > tempfile.txt
[Joe@file-integrity ~]$ ls
tempfile.txt
[Joe@file-integrity ~]$ cat tempfile.ttx
cat: tempfile.ttx: No such file or directory
[Joe@file-integrity ~]$ cat tempfile.txt
tempfile content
[Joe@file-integrity ~]$ sha1sum tempfile.txt
6898e77845a72af1e220d8e5b93f01aa7f8a11b8  tempfile.txt
[Joe@file-integrity ~]$ ls
tempfile.txt
[Joe@file-integrity ~]$ █
```

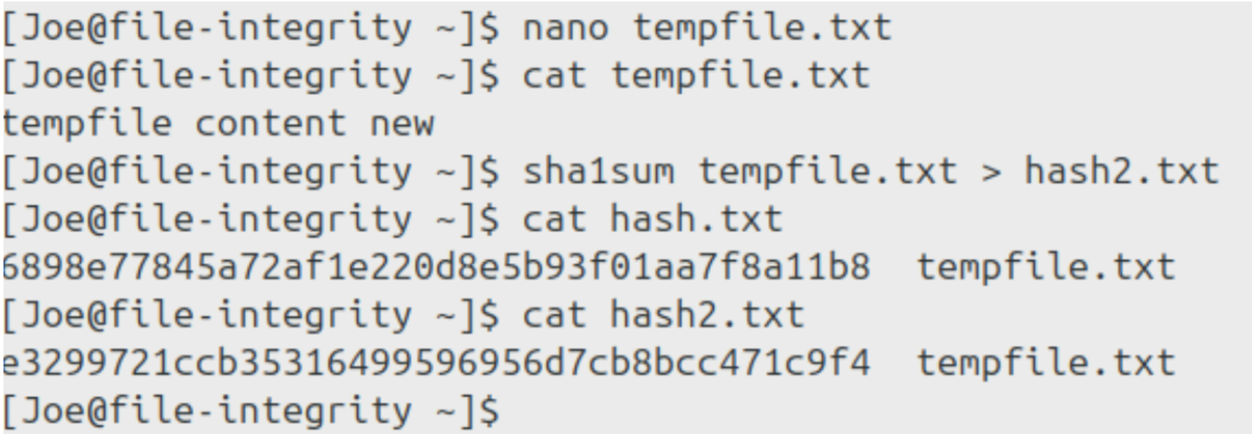
Basically, I'm creating a temp file adding some text using echo, checking the file, hashing the file using sha1sum and checking it once again.
B



```
[Joe@file-integrity ~]$ sha1sum tempfile.txt > hash.txt
[Joe@file-integrity ~]$ ls
hash.txt  tempfile.txt
[Joe@file-integrity ~]$ hash.txt
-bash: hash.txt: command not found
[Joe@file-integrity ~]$ cat hash.txt
6898e77845a72af1e220d8e5b93f01aa7f8a11b8  tempfile.txt
[Joe@file-integrity ~]$ █
```

Kinda what I did in A but specifically about the hashing and checking part.

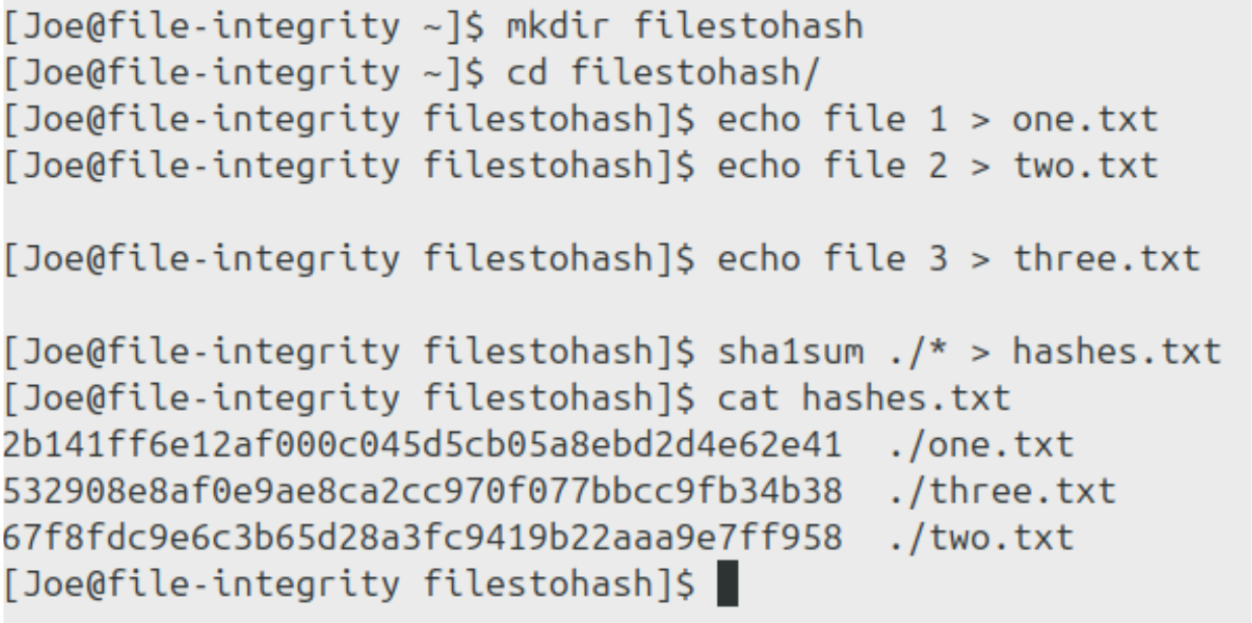
C



```
[Joe@file-integrity ~]$ nano tempfile.txt
[Joe@file-integrity ~]$ cat tempfile.txt
tempfile content new
[Joe@file-integrity ~]$ sha1sum tempfile.txt > hash2.txt
[Joe@file-integrity ~]$ cat hash.txt
6898e77845a72af1e220d8e5b93f01aa7f8a11b8  tempfile.txt
[Joe@file-integrity ~]$ cat hash2.txt
e3299721ccb35316499596956d7cb8bcc471c9f4  tempfile.txt
[Joe@file-integrity ~]$
```

I end up editing the text of the temp file, check it, hash it again and place it into another file so I can compare the two. NOTE: The outputs of each hash are completely different

D



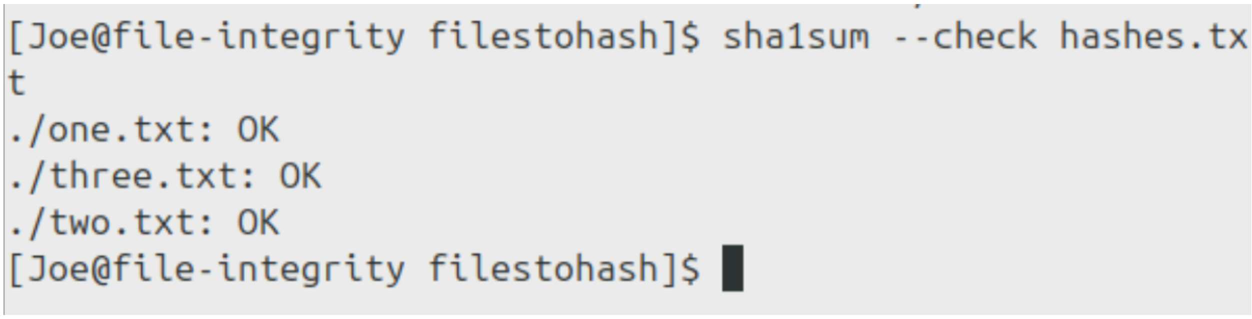
```
[Joe@file-integrity ~]$ mkdir filestohash
[Joe@file-integrity ~]$ cd filestohash/
[Joe@file-integrity filestohash]$ echo file 1 > one.txt
[Joe@file-integrity filestohash]$ echo file 2 > two.txt

[Joe@file-integrity filestohash]$ echo file 3 > three.txt

[Joe@file-integrity filestohash]$ sha1sum ./* > hashes.txt
[Joe@file-integrity filestohash]$ cat hashes.txt
2b141ff6e12af000c045d5cb05a8ebd2d4e62e41  ./one.txt
532908e8af0e9ae8ca2cc970f077bbcc9fb34b38  ./three.txt
67f8fdc9e6c3b65d28a3fc9419b22aaa9e7ff958  ./two.txt
[Joe@file-integrity filestohash]$
```

I create a folder called filestohash containing 3 files, one, two and three, move to said folder and then hash each of the files, putting the result in hashes.txt

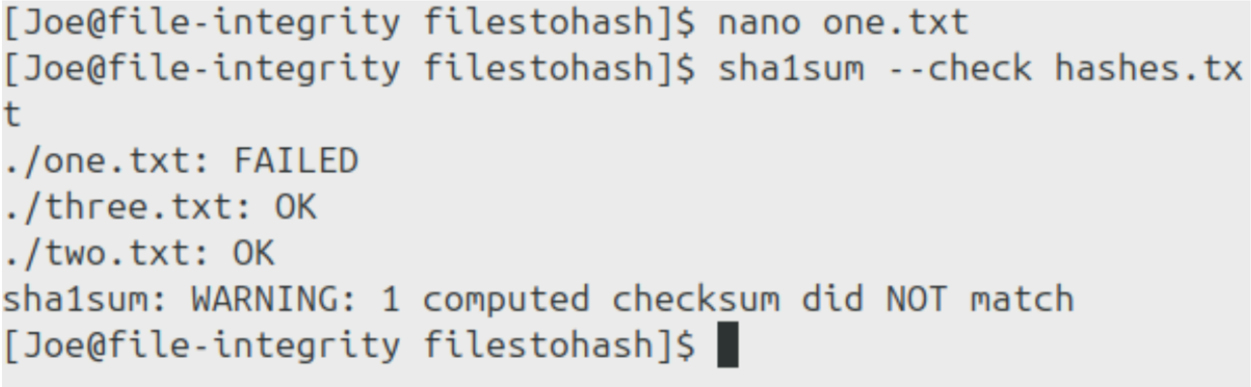
E



```
[Joe@file-integrity filestohash]$ sha1sum --check hashes.tx
t
./one.txt: OK
./three.txt: OK
./two.txt: OK
[Joe@file-integrity filestohash]$
```

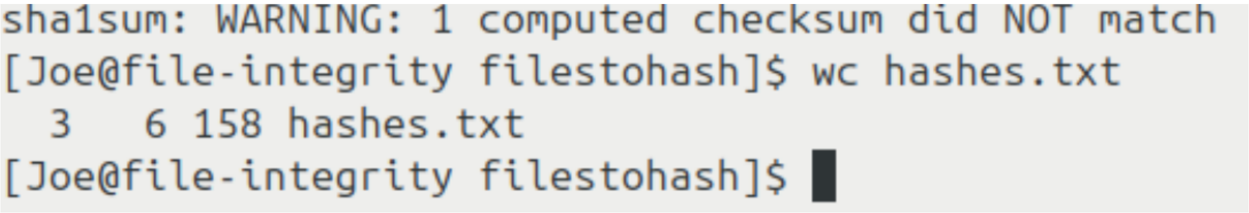
I check the integrity of the hashes using –check.

F

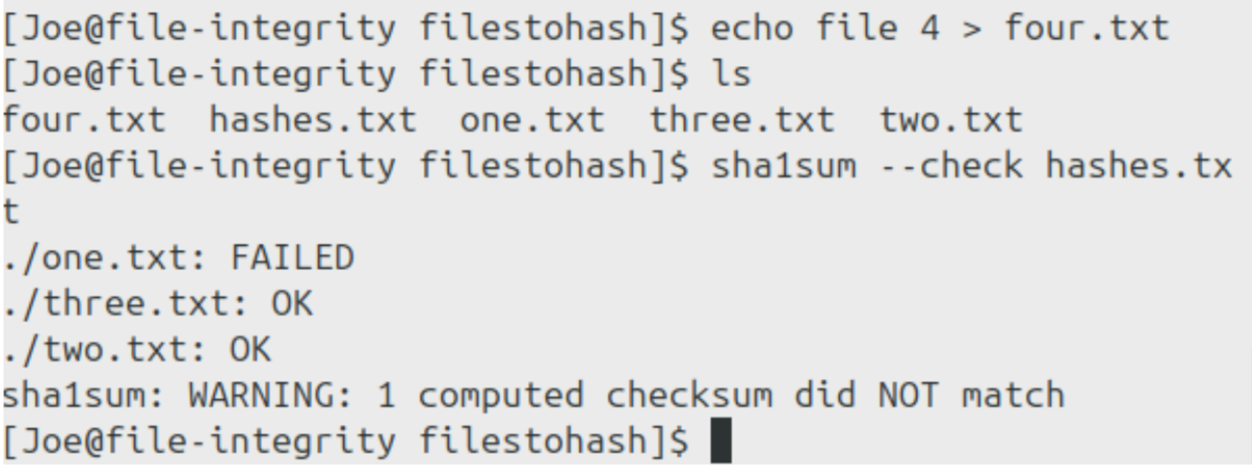


```
[Joe@file-integrity filestohash]$ nano one.txt
[Joe@file-integrity filestohash]$ sha1sum --check hashes.tx
t
./one.txt: FAILED
./three.txt: OK
./two.txt: OK
sha1sum: WARNING: 1 computed checksum did NOT match
[Joe@file-integrity filestohash]$
```

I edit one of the files and check if there are any changes to the integrity of the files.
G



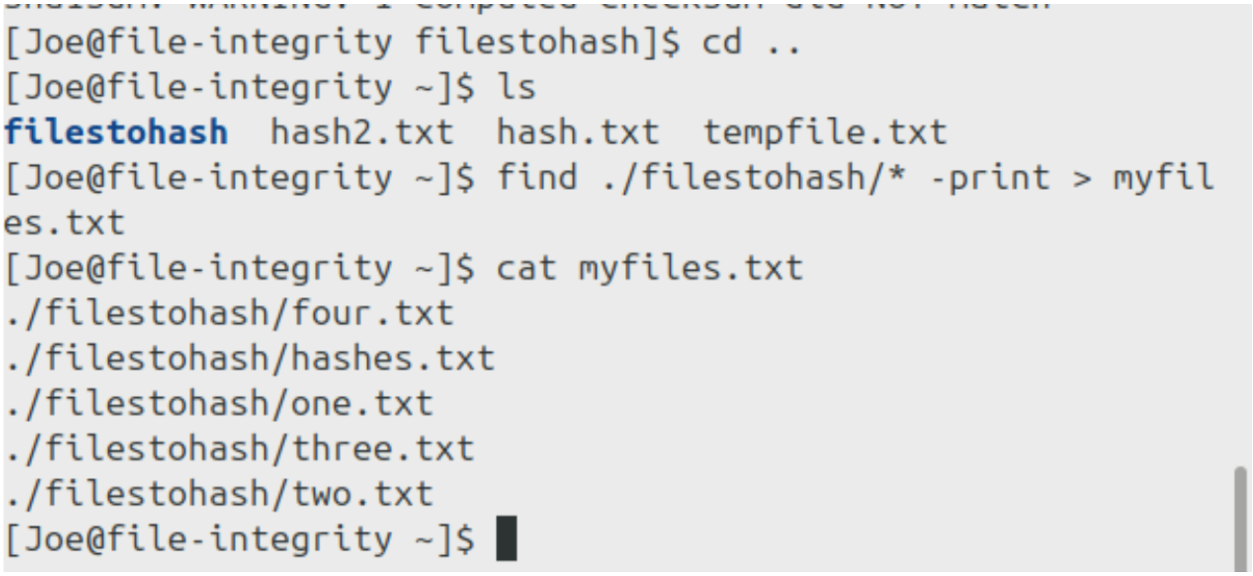
```
sha1sum: WARNING: 1 computed checksum did NOT match
[Joe@file-integrity filestohash]$ wc hashes.txt
  3   6 158 hashes.txt
[Joe@file-integrity filestohash]$
```



```
[Joe@file-integrity filestohash]$ echo file 4 > four.txt
[Joe@file-integrity filestohash]$ ls
four.txt  hashes.txt  one.txt  three.txt  two.txt
[Joe@file-integrity filestohash]$ sha1sum --check hashes.tx
t
./one.txt: FAILED
./three.txt: OK
./two.txt: OK
sha1sum: WARNING: 1 computed checksum did NOT match
[Joe@file-integrity filestohash]$
```

I count the number of files hashed using the command wc and then I check the integrity of the files after I created a new file called four.

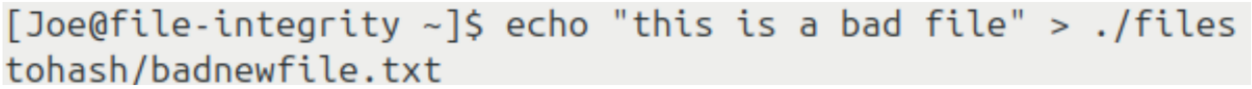
H



```
sha1sum: WARNING: 1 computed checksum did NOT match
[Joe@file-integrity filestohash]$ cd ..
[Joe@file-integrity ~]$ ls
filestohash  hash2.txt  hash.txt  tempfile.txt
[Joe@file-integrity ~]$ find ./filestohash/* -print > myfil
es.txt
[Joe@file-integrity ~]$ cat myfiles.txt
./filestohash/four.txt
./filestohash/hashes.txt
./filestohash/one.txt
./filestohash/three.txt
./filestohash/two.txt
[Joe@file-integrity ~]$ 
```

I used the find command to create a file containing the list of every file in the filestohash folder

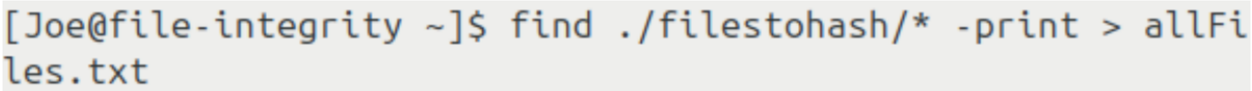
I



```
[Joe@file-integrity ~]$ echo "this is a bad file" > ./files
tohash/badnewfile.txt
```

I create a new file, adding it to the filestohash folder

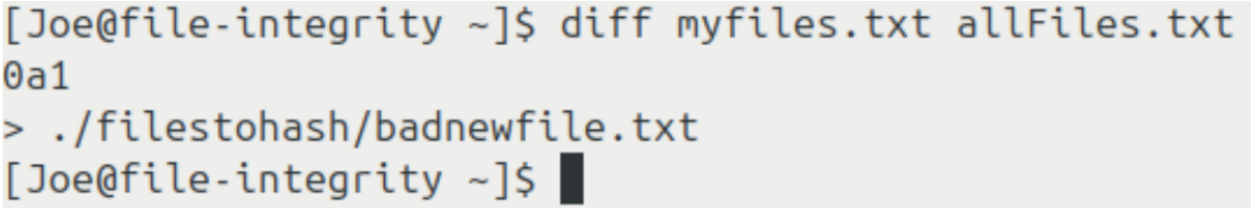
J



```
[Joe@file-integrity ~]$ find ./filestohash/* -print > allFi
les.txt
```

I then redo what I did from H but add it into another file

K

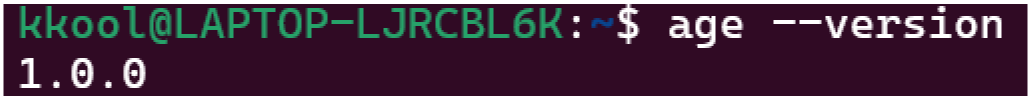


```
[Joe@file-integrity ~]$ diff myfiles.txt allFiles.txt
0a1
> ./filestohash/badnewfile.txt
[Joe@file-integrity ~]$ 
```

Finally, I compare the difference between files.

Task 2

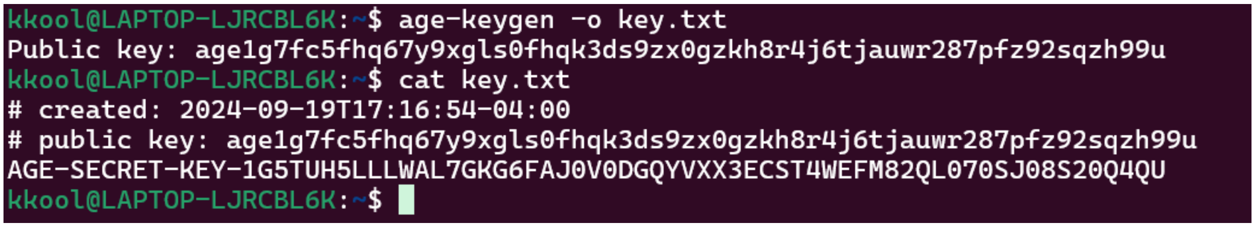
A



```
kkool@LAPTOP-LJRCBL6K:~$ age --version
1.0.0
```

Confirmation of installing age

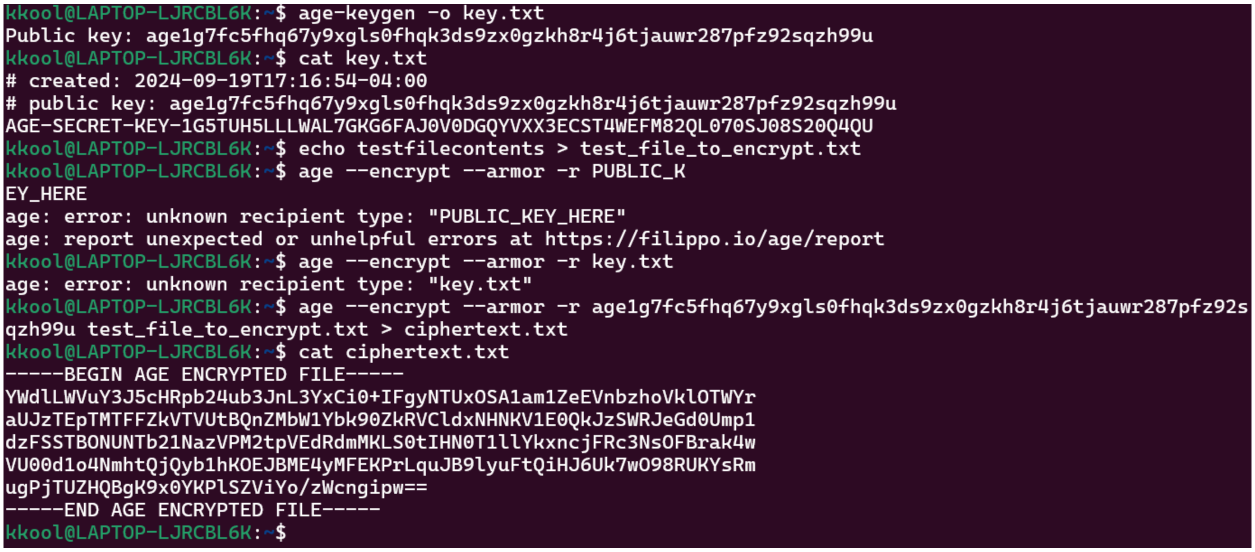
B



```
kkool@LAPTOP-LJRCBL6K:~$ age-keygen -o key.txt
Public key: age1g7fc5fhq67y9xgls0fhqk3ds9zx0gzkh8r4j6tjauwr287pfz92sqzh99u
kkool@LAPTOP-LJRCBL6K:~$ cat key.txt
# created: 2024-09-19T17:16:54-04:00
# public key: age1g7fc5fhq67y9xgls0fhqk3ds9zx0gzkh8r4j6tjauwr287pfz92sqzh99u
AGE-SECRET-KEY-1G5TUH5LLLWAL7GKG6FAJ0V0DGQYVXX3ECST4WEFM82QL070SJ08S20Q4QU
kkool@LAPTOP-LJRCBL6K:~$ 
```

I created a public key, stored it into key.txt and observed it using cat

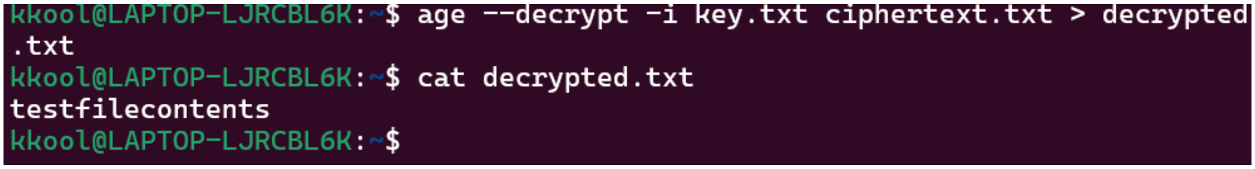
C & D



```
kkool@LAPTOP-LJRCBL6K:~$ age-keygen -o key.txt
Public key: age1g7fc5fhq67y9xgls0fhqk3ds9zx0gzkh8r4j6tjauwr287pfz92sqzh99u
kkool@LAPTOP-LJRCBL6K:~$ cat key.txt
# created: 2024-09-19T17:16:54-04:00
# public key: age1g7fc5fhq67y9xgls0fhqk3ds9zx0gzkh8r4j6tjauwr287pfz92sqzh99u
AGE-SECRET-KEY-1G5TUH5LLLWAL7GKG6FAJ0V0DGQYVXX3ECST4WEFM82QL070SJ08S20Q4QU
kkool@LAPTOP-LJRCBL6K:~$ echo testfilecontents > test_file_to_encrypt.txt
kkool@LAPTOP-LJRCBL6K:~$ age --encrypt --armor -r PUBLIC_K
EY_HERE
age: error: unknown recipient type: "PUBLIC_KEY_HERE"
age: report unexpected or unhelpful errors at https://filippo.io/age/report
kkool@LAPTOP-LJRCBL6K:~$ age --encrypt --armor -r key.txt
age: error: unknown recipient type: "key.txt"
kkool@LAPTOP-LJRCBL6K:~$ age --encrypt --armor -r age1g7fc5fhq67y9xgls0fhqk3ds9zx0gzkh8r4j6tjauwr287pfz92s
qzh99u test_file_to_encrypt.txt > ciphertext.txt
kkool@LAPTOP-LJRCBL6K:~$ cat ciphertext.txt
-----BEGIN AGE ENCRYPTED FILE-----
YWdlLWVuY3J5cHRpb24ub3JnL3YxCi0+IFgyNTUxOSA1am1ZeEVnbzhoVklOTWYr
aUJzTEpTMTFFZkVTVUtBQnZMbW1Ybk90ZkRVCldxNHNKV1E0QkJzSWRJeGd0Ump1
dzFSSTBONUNTb21NazVPM2tpVEdRdmMKLS0tIHN0T1llYkxncjFRc3NsOFBrak4w
VU00d1o4NmhtQjQyb1hKOEJBME4yMFEKPrLquJB9lyuFtQiHJ6Uk7wO98RUKYsRm
ugPjTUZHQBgK9x0YKPlSZViYo/zWcngipw==
-----END AGE ENCRYPTED FILE-----
kkool@LAPTOP-LJRCBL6K:~$
```

Next, I created a file, called test_file_to_encrypt.txt, encrypted it using my public key and then I viewed it.

E

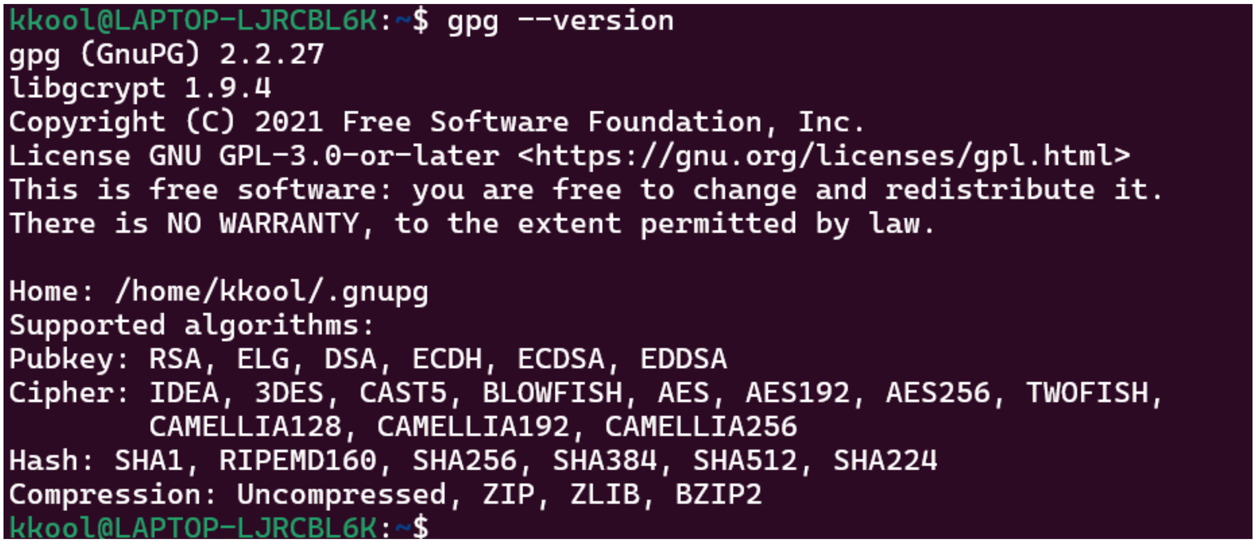


```
kkool@LAPTOP-LJRCBL6K:~$ age --decrypt -i key.txt ciphertext.txt > decrypted
.txt
kkool@LAPTOP-LJRCBL6K:~$ cat decrypted.txt
testfilecontents
kkool@LAPTOP-LJRCBL6K:~$
```

Finally I decrypted the file

Task 3

A

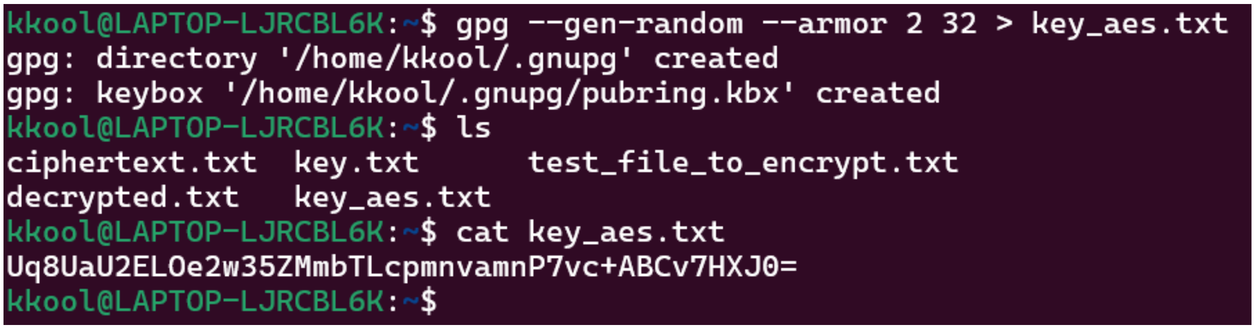


```
kkool@LAPTOP-LJRCBL6K:~$ gpg --version
gpg (GnuPG) 2.2.27
libgcrypt 1.9.4
Copyright (C) 2021 Free Software Foundation, Inc.
License GNU GPL-3.0-or-later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: /home/kkool/.gnupg
Supported algorithms:
Pubkey: RSA, ELG, DSA, ECDH, ECDSA, EDDSA
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
        CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2
kkool@LAPTOP-LJRCBL6K:~$
```

Confirmation of gpg install

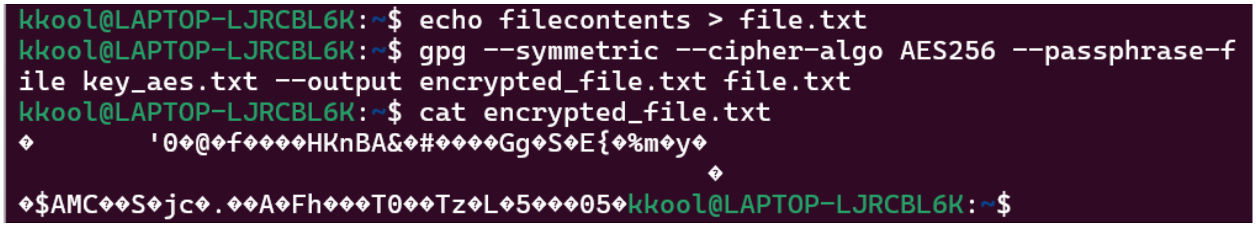
B



```
kkool@LAPTOP-LJRCBL6K:~$ gpg --gen-random --armor 2 32 > key_aes.txt
gpg: directory '/home/kkool/.gnupg' created
gpg: keybox '/home/kkool/.gnupg/pubring.kbx' created
kkool@LAPTOP-LJRCBL6K:~$ ls
ciphertext.txt  key.txt      test_file_to_encrypt.txt
decrypted.txt   key_aes.txt
kkool@LAPTOP-LJRCBL6K:~$ cat key_aes.txt
Uq8UaU2ELOe2w35ZMmbTLcpmnvamnP7vc+ABCv7HXJ0=
kkool@LAPTOP-LJRCBL6K:~$
```

Created a 256 bit AES key and checked the contents of the file I stored it in.

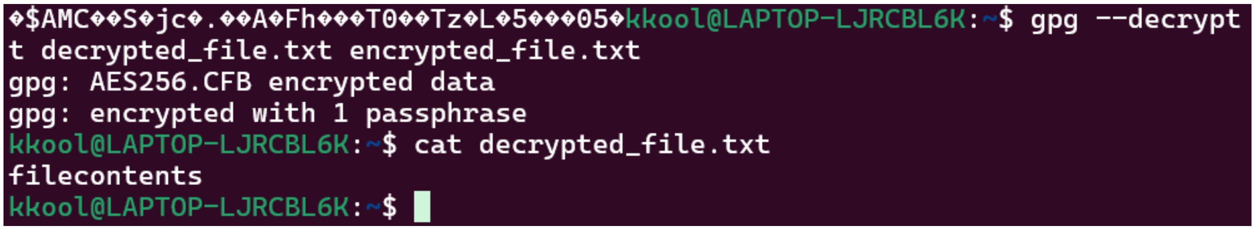
C



```
kkool@LAPTOP-LJRCBL6K:~$ echo filecontents > file.txt
kkool@LAPTOP-LJRCBL6K:~$ gpg --symmetric --cipher-algo AES256 --passphrase-f
ile key_aes.txt --output encrypted_file.txt file.txt
kkool@LAPTOP-LJRCBL6K:~$ cat encrypted_file.txt
�       '0��@�f����HKnBA&�#����Gg�S�E{�%m�y�
                                             �
�$AMC��S�jc�.��A�Fh���T0��Tz�L�5���05�kkool@LAPTOP-LJRCBL6K:~$
```

I created a new file, encrypted it using my AES key and then observed it.

D



```
�$AMC��S�jc�.��A�Fh���T0��Tz�L�5���05�kkool@LAPTOP-LJRCBL6K:~$ gpg --decrypt
t decrypted_file.txt encrypted_file.txt
gpg: AES256.CFB encrypted data
gpg: encrypted with 1 passphrase
kkool@LAPTOP-LJRCBL6K:~$ cat decrypted_file.txt
filecontents
kkool@LAPTOP-LJRCBL6K:~$
```

Afterwards, I decrypted it.