TASK 1: You are assigned with two files (file1.txt, Challange.txt) on each of them and you have to apply a fixed length encoding scheme and show its performance in form of time complexity, size of file.

For this task, we searched up various fixed-length algorithms online and in textbooks. We shortlisted various algorithms such as AES Encryption, SHA-1 technique etc. We took into consideration many factors including time constraints, workload etc. and we decided to move forward with AES Encoding technique.

AES Encoding Algorithm

AES stands for Advanced Encryption Standard. It is mostly used as a replacement of DES. It has a bigger key size than DES which provides encryption at least 6 times faster. AES specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information. Encryption converts data to an unintelligible form called ciphertext; decrypting the ciphertext converts the data back into its original form, called plaintext. The AES algorithm can use cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits. It has many advantages and disadvantages. Advantages include it being secure and used globally. But it has certain disadvantages too which we will see in our test results.

In our implementation, we are using a single file named **"aes.cpp"**. We implemented both encryption and decryption in C++. Once the file is encrypted, it is stored in **"encrypted.txt"** and similarly the contents that are written in encoded.txt are then decrypted and written in **"decrypted.txt"**.

We then performed tests using two of test files named ***"file1.txt"*** and ***"Challange.txt"***. Following is the results after running the tests. *(For most accurate results, we ran each file 3 times and calculated its average and store it)*

To run the file, follow the following steps:

- g++ aes.cpp
- ./a.out

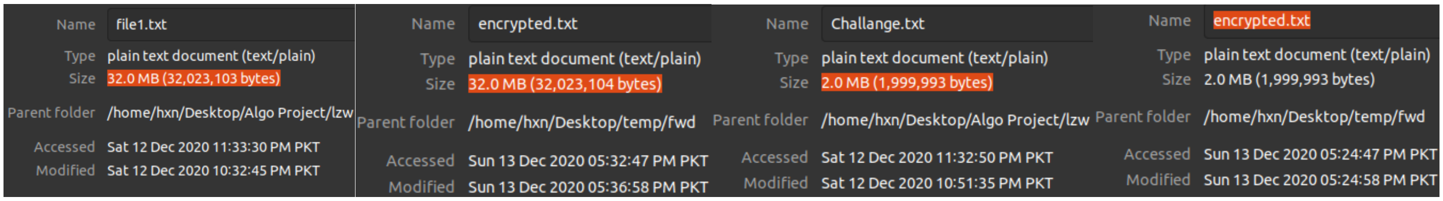
It will then ask you to enter the name of the file. Enter the name **without the extension (i.e., ".txt")**. Enter the name of the file you want to encrypt.

Results:

The results of the tests are as follows:

Name of the file	Original Size (MBs)	Size after Encryption (MBs)	Decrease %	Time for Encryption
“file1.txt”	32.0 (32,023,103B)	32.0 (32,023,103B)	0%	186.7s (3.1 mins)
“Challange.txt”	2.0 (1,999,993B)	2.0 (1,999,993B)	0%	11.4s (0.19 mins)

Note: The “Time of Encryption” includes the time taken to read the test file.



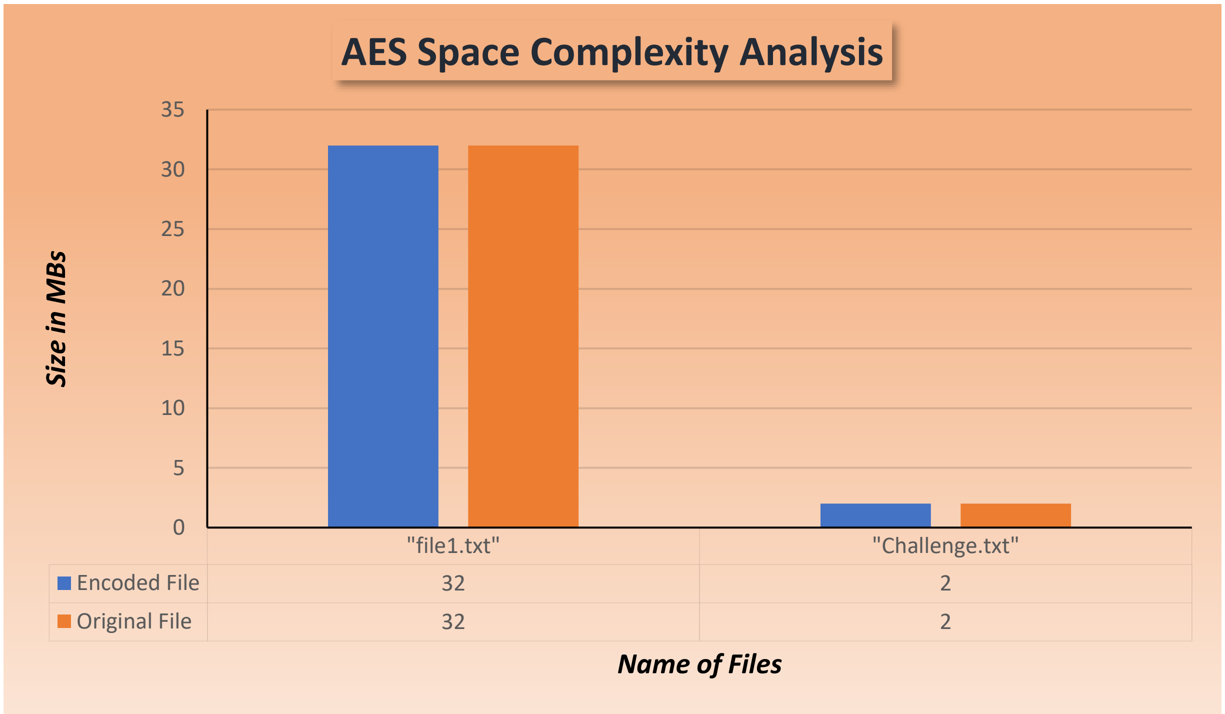
Name file1.txt
Type plain text document (text/plain)
Size 32.0 MB (32,023,103 bytes)
Parent folder /home/hxn/Desktop/Algo Project/lzw
Accessed Sat 12 Dec 2020 11:33:30 PM PKT
Modified Sat 12 Dec 2020 10:32:45 PM PKT

Name encrypted.txt
Type plain text document (text/plain)
Size 32.0 MB (32,023,104 bytes)
Parent folder /home/hxn/Desktop/temp/fwd
Accessed Sun 13 Dec 2020 05:32:47 PM PKT
Modified Sun 13 Dec 2020 05:36:58 PM PKT

Name Challange.txt
Type plain text document (text/plain)
Size 2.0 MB (1,999,993 bytes)
Parent folder /home/hxn/Desktop/Algo Project/lzw
Accessed Sat 12 Dec 2020 11:32:50 PM PKT
Modified Sat 12 Dec 2020 10:51:35 PM PKT

Name encrypted.txt
Type plain text document (text/plain)
Size 2.0 MB (1,999,993 bytes)
Parent folder /home/hxn/Desktop/temp/fwd
Accessed Sun 13 Dec 2020 05:24:47 PM PKT
Modified Sun 13 Dec 2020 05:24:58 PM PKT

AES Encoding Algorithm Benchmarks:



	"file1.txt"	"Challenge.txt"
■ Encoded File	32	2
■ Original File	32	2

As can be seen from the graph, the disadvantage of the AES Encoding algorithm is that it is fixed length encoding scheme but it does not compress a file. This is because this scheme generates encryption keys randomly due to which the size varies. Moreover, this algorithm doesn’t reduce the size of the file as it is not a data compression algorithm.