# Project

For this project, I had to investigate whether two files were identical or different. My task involved several steps to accomplish this goal. First, I needed to display the contents of both files to visually inspect their data. Then, I had to create hash values for each file using a hashing algorithm. This process would generate unique digital fingerprints for the files based on their contents. After obtaining the hashes, my next step was to carefully examine these hash values. Finally, I had to compare the hashes of the two files. If the hash values matched, it would indicate that the files were identical in content, despite potentially having different names or locations. If the hashes differed, it would confirm that the files had distinct contents. This method provided a reliable and efficient way to determine file similarity or difference without relying solely on visual comparison of the file contents.

I used the `ls` command to list the contents of the directory.

```
st@c8179a00a6d2:-$ ls
.txt file2.txt
```

There were two files listed: `file1.txt` and `file2.txt`. I used the cat command to display the contents of the `file1.txt` file. Then, I used the cat command again to display the contents of the `file2.txt` file.

```
analyst@c8179a00a6d2:~$ cat file1.txt
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
analyst@c8179a00a6d2:~$ cat file2.txt
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

I used the `sha256sum` command to generate the hash of the `file1.txt` file. Then, I used the `sha256sum` command to generate the hash of the `file2.txt` file.

```
9sxa5Yq20Ranalyst@c8179a00a6d2:~$ sha256sum file1.txt
131f95c51cc819465fa1797f6ccacf9d494aaaff46fa3eac73ae63ffbdfd8267  file1.txt
analyst@c8179a00a6d2:~$ sha256sum file2.txt
2558ba9a4cad1e69804ce03aa2a029526179a91a5e38cb723320e83af9ca017b  file2.txt
```

I used the `sha256sum` command to generate the hash of the `file1.txt` file and sent the output to a new file called `file1hash`. Then, I used the `sha256sum` command to generate the hash of the `file2.txt` file and sent the output to a new file called `file2hash`.

```
analyst@c8179a00a6d2:~$ sha256sum file1.txt >> file1hash
analyst@c8179a00a6d2:~$ sha256sum file2.txt >> file2hash
analyst@c8179a00a6d2:~$ cat file1hash
131f95c51cc819465fa1797f6ccacf9d494aaaff46fa3eac73ae63ffbdfd8267  file1.txt
analyst@c8179a00a6d2:~$ cat file2hash
2558ba9a4cad1e69804ce03aa2a029526179a91a5e38cb723320e83af9ca017b  file2.txt
```