```
Session  Actions  Edit  View  Help

┌──(anki㉿kali)-[~/Exp2]
└─$ nano file.txt


┌──(anki㉿kali)-[~/Exp2]
└─$ cp file.txt filecopy1.txt


┌──(anki㉿kali)-[~/Exp2]
└─$ ls
filecopy1.txt  file.txt

┌──(anki㉿kali)-[~/Exp2]
└─$ nano sha256.py


┌──(anki㉿kali)-[~/Exp2]
└─$ python3 sha256.py

═══ SHA-256 Hashing Menu ═══
1. Generate SHA-256 Hash of a File
2. Verify Integrity (Compare Two Files)
3. Exit
Enter your choice (1-3): 1
Enter file name: file.txt

SHA-256 Hash: 67c5a205a78d68a48236309b261ed6d387450bfbd6faa4295ba233a0024343bc

═══ SHA-256 Hashing Menu ═══
1. Generate SHA-256 Hash of a File
2. Verify Integrity (Compare Two Files)
3. Exit
Enter your choice (1-3): 1
Enter file name: filecopy1.txt

SHA-256 Hash: 67c5a205a78d68a48236309b261ed6d387450bfbd6faa4295ba233a0024343bc

═══ SHA-256 Hashing Menu ═══
1. Generate SHA-256 Hash of a File
2. Verify Integrity (Compare Two Files)
3. Exit
Enter your choice (1-3): 2
Enter ORIGINAL file name: file.txt
Enter COPIED/MODIFIED file name: filecopy1.txt

Hash of file.txt : 67c5a205a78d68a48236309b261ed6d387450bfbd6faa4295ba233a0024343bc
Hash of filecopy1.txt : 67c5a205a78d68a48236309b261ed6d387450bfbd6faa4295ba233a0024343bc

Integrity Verified  (Files are same)

═══ SHA-256 Hashing Menu ═══
1. Generate SHA-256 Hash of a File
2. Verify Integrity (Compare Two Files)
3. Exit
Enter your choice (1-3):
```

```
  GNU nano 8.7
import hashlib
import os

def sha256_hash(filename):
    with open(filename, "rb") as f:
        data = f.read()
    return hashlib.sha256(data).hexdigest()

def verify_integrity(file1, file2):
    hash1 = sha256_hash(file1)
    hash2 = sha256_hash(file2)
    return hash1, hash2, (hash1 == hash2)


while True:
    print("\n==== SHA-256 Hashing Menu ====")
    print("1. Generate SHA-256 Hash of a File")
    print("2. Verify Integrity (Compare Two Files)")
    print("3. Exit")

    choice = input("Enter your choice (1-3): ")

    if choice == "1":
        filename = input("Enter file name: ")

        if not os.path.exists(filename):
            print("File does not exist!")
            continue

        digest = sha256_hash(filename)
        print("\nSHA-256 Hash:", digest)

    elif choice == "2":
        file1 = input("Enter ORIGINAL file name: ")
        file2 = input("Enter COPIED/MODIFIED file name: ")

        if not (os.path.exists(file1) and os.path.exists(file2)):
            print("One or both files do not exist!")
            continue

        h1, h2, match = verify_integrity(file1, file2)

        print("\nHash of", file1, ":", h1)
        print("Hash of", file2, ":", h2)

        if match:
            print("\nIntegrity Verified  (Files are same)")
        else:
            print("\nIntegrity Failed (Files are different)")

    elif choice == "3":
        print("Exiting")
        break

    else:
        print("Invalid choice. Please enter 1-3 only.")


^G Help          ^O Write Out     ^F Where Is      ^K Cut           ^T Execute
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify
```