EX.NO.: 05
DATE: 09.07.2025

GENERATING AND VERIFYING MD5 AND SHA256 HASHES FOR STRINGS

AIM

To generate and verify MD5 and SHA-256 hashes for input strings using Python's `hashlib` library, ensuring data integrity and authenticity.

ALGORITHM

1. Accept an input string from the user.
2. Generate the MD5 hash of the string using `hashlib.md5()`.
3. Generate the SHA-256 hash of the string using `hashlib.sha256()`.
4. Display both hash values in hexadecimal format.
5. Accept a known hash value for verification.
6. Determine the hashing algorithm based on the hash length:
7. 32 characters → MD5
8. 64 characters → SHA-256
9. Recompute the hash using the determined algorithm.
10. Compare the computed hash with the known hash.
11. Display whether verification was successful (`True`) or not (`False`).

CODE AND OUTPUT

```
import hashlib

def generate_hashes(input_string):
    md5_hash = hashlib.md5(input_string.encode()).hexdigest()
    sha256_hash = hashlib.sha256(input_string.encode()).hexdigest()
    return md5_hash, sha256_hash

def verify_hash(input_string, known_hash):
    if len(known_hash) == 32:
        computed_hash = hashlib.md5(input_string.encode()).hexdigest()
        algorithm = "MD5"
    elif len(known_hash) == 64:
        computed_hash = hashlib.sha256(input_string.encode()).hexdigest()
        algorithm = "SHA-256"
    else:
        print("Unknown hash length. Cannot verify.")
        return False

    match = computed_hash == known_hash
    print(f"Verifying using {algorithm}: {match}")
    return match

# ---- Example Usage ----
if __name__ == "__main__":
    input_str = input("Enter a string to hash: ")

    md5_result, sha256_result = generate_hashes(input_str)
    print(f"\nMD5 Hash:    {md5_result}")
```

```
    print(f"SHA-256 Hash: {sha256_result}")

    # Verification
    known = input("\nEnter a known hash to verify against the input string: ")
    verify_hash(input_str, known)
```

```
MD5 Hash:    5d41402abc4b2a76b9719d911017c592
SHA-256 Hash: 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
Verifying using MD5: True
```

INFERENCE

The script successfully generates and verifies both MD5 and SHA-256 hashes, demonstrating how hashing algorithms ensure input integrity and secure verification without storing the original data.