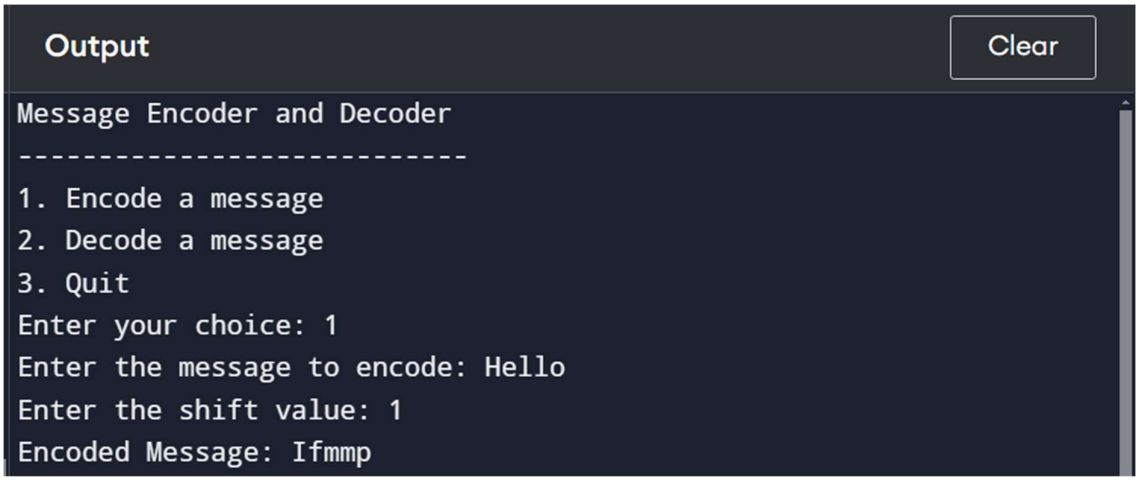
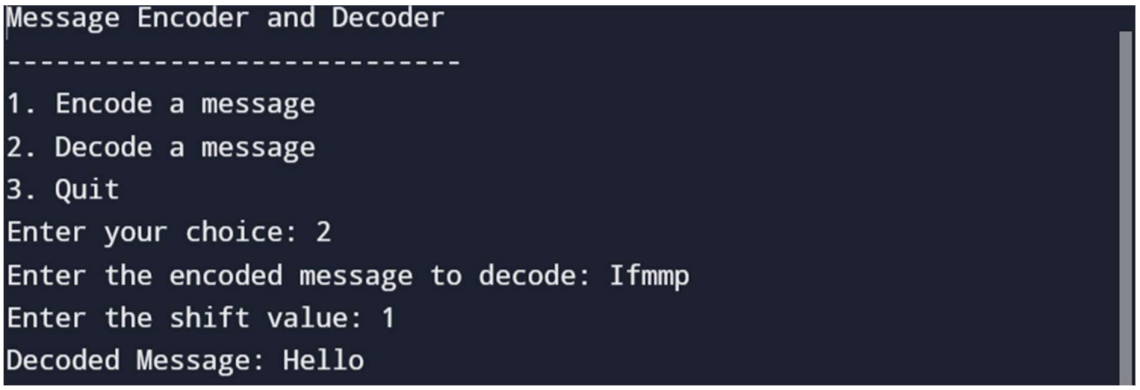
Encode-Decode.py

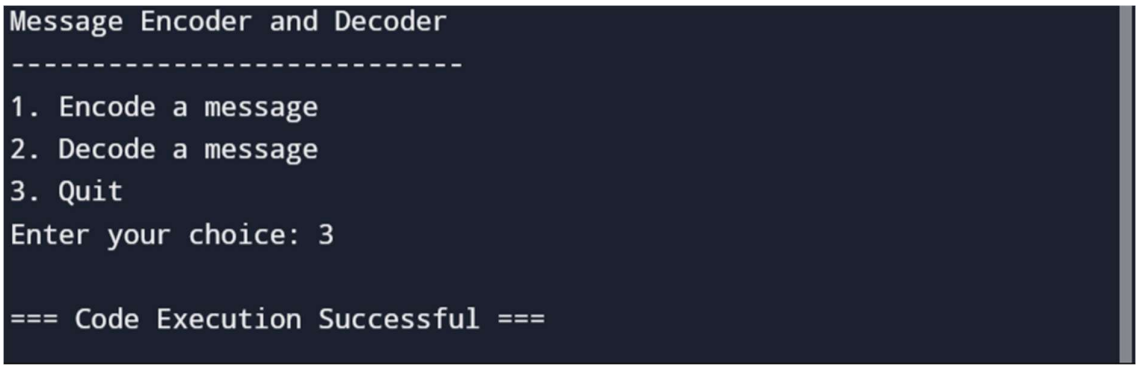
```python
class Encoder:
    def __init__(self, shift):
        """
        Initialize the Encoder class with a shift value.

        Args:
            shift (int): The number of positions to shift the characters.
        """
        self.shift = shift

    def encode(self, message):
        """
        Encode the message using the Caesar Cipher algorithm.

        Args:
            message (str): The message to be encoded.

        Returns:
            str: The encoded message.
        """
        encoded_message = ""
        for char in message:
            if char.isalpha():
                ascii_offset = 65 if char.isupper() else 97
                encoded_char = chr((ord(char) - ascii_offset + self.shift) % 26 +
ascii_offset)
                encoded_message += encoded_char
            else:
                encoded_message += char
        return encoded_message


class Decoder:
    def __init__(self, shift):
        """
        Initialize the Decoder class with a shift value.

        Args:
            shift (int): The number of positions to shift the characters.
        """
        self.shift = shift

    def decode(self, encoded_message):
        """
        Decode the encoded message using the Caesar Cipher algorithm.

        Args:
            encoded_message (str): The encoded message to be decoded.

        Returns:
            str: The decoded message.
        """
```

```
        decoded_message = ""
        for char in encoded_message:
            if char.isalpha():
                ascii_offset = 65 if char.isupper() else 97
                decoded_char = chr((ord(char) - ascii_offset - self.shift) % 26 +
ascii_offset)
                decoded_message += decoded_char
            else:
                decoded_message += char
        return decoded_message


def main():
    while True:
        print("Message Encoder and Decoder")
        print("----------------------------")
        print("1. Encode a message")
        print("2. Decode a message")
        print("3. Quit")
        choice = input("Enter your choice: ")

        if choice == "1":
            message = input("Enter the message to encode: ")
            shift = int(input("Enter the shift value: "))
            encoder = Encoder(shift)
            encoded_message = encoder.encode(message)
            print(f"Encoded Message: {encoded_message}")
        elif choice == "2":
            encoded_message = input("Enter the encoded message to decode: ")
            shift = int(input("Enter the shift value: "))
            decoder = Decoder(shift)
            decoded_message = decoder.decode(encoded_message)
            print(f"Decoded Message: {decoded_message}")
        elif choice == "3":
            break
        else:
            print("Invalid choice. Please try again.")


if __name__ == "__main__":
    main()
```

OUTPUT


Output
Clear
Message Encoder and Decoder
---
1. Encode a message
2. Decode a message
3. Quit
Enter your choice: 1
Enter the message to encode: Hello
Enter the shift value: 1
Encoded Message: Ifmmp


Message Encoder and Decoder
---
1. Encode a message
2. Decode a message
3. Quit
Enter your choice: 2
Enter the encoded message to decode: Ifmmp
Enter the shift value: 1
Decoded Message: Hello


Message Encoder and Decoder
---
1. Encode a message
2. Decode a message
3. Quit
Enter your choice: 3
=== Code Execution Successful ===