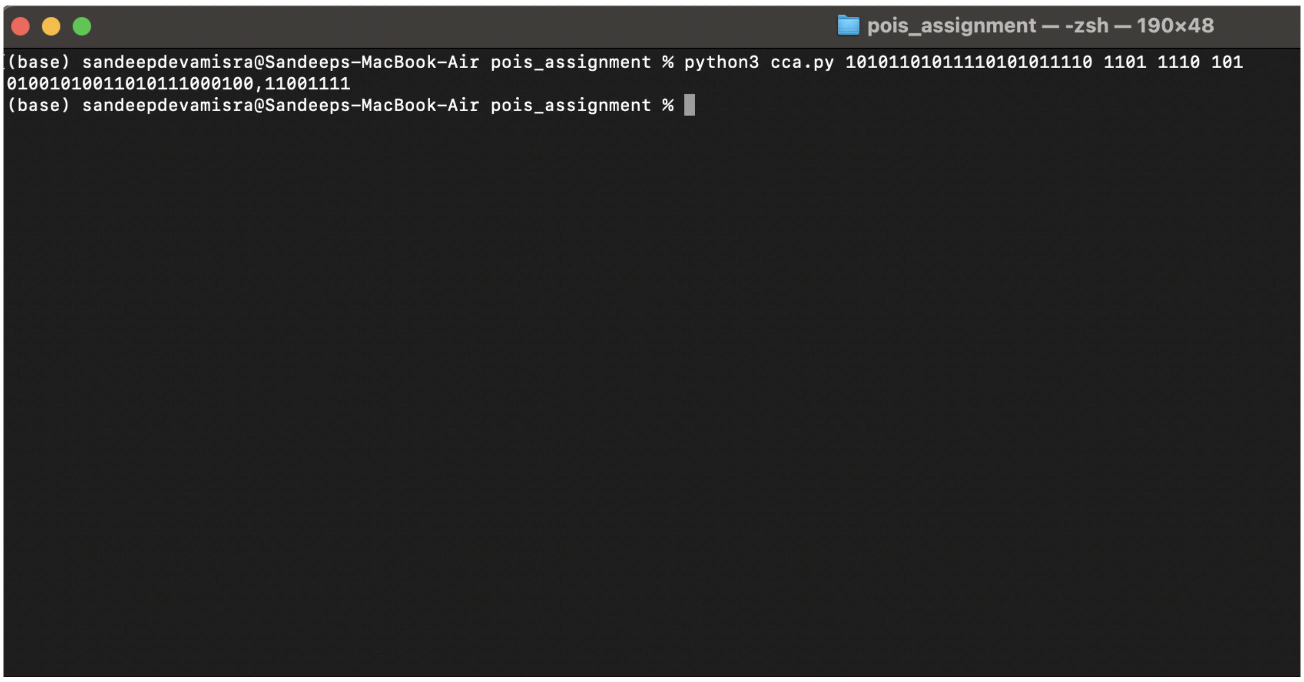
Filename: cca.py
Running instruction: `python3 cca.py <message> <seed1> <seed2> <counter>`
Input: message, seed1, seed2, counter
Class: CCA
Function: encrypt, decrypt, handler

The CPA and MAC classes are imported and used here directly. The input message is preprocessed in the constructor itself since the format of the message string will decide whether to encrypt or decrypt. Input string without a comma-delimiter suggests that it is a plaintext and needs to be encrypted. If it is comma-delimited then the string is a <ciphertext and tag> instead and therefore needs to be decrypted. The encrypt function uses the CPA object to encrypt the plaintext and this encrypted plaintext will be given to the MAC object to generate the tag. Both the encrypted plaintext and the tag are concatenated by taking a comma as a delimiter and a single string is returned as output.
In case of the decrypt function, the MAC object is used to again generate a tag for the ciphertext. This tag is compared with the tag that was obtained. If they match, then the CPA object is used to decrypt the ciphertext.

The handler function simply checks whether to encrypt or decrypt based on the format of input.



```
(base) sandeepdevamisra@Sandeeps-MacBook-Air pois_assignment % python3 cca.py 10101101011110101011110 1101 1110 101
01001010011010111000100,11001111
(base) sandeepdevamisra@Sandeeps-MacBook-Air pois_assignment % 
```