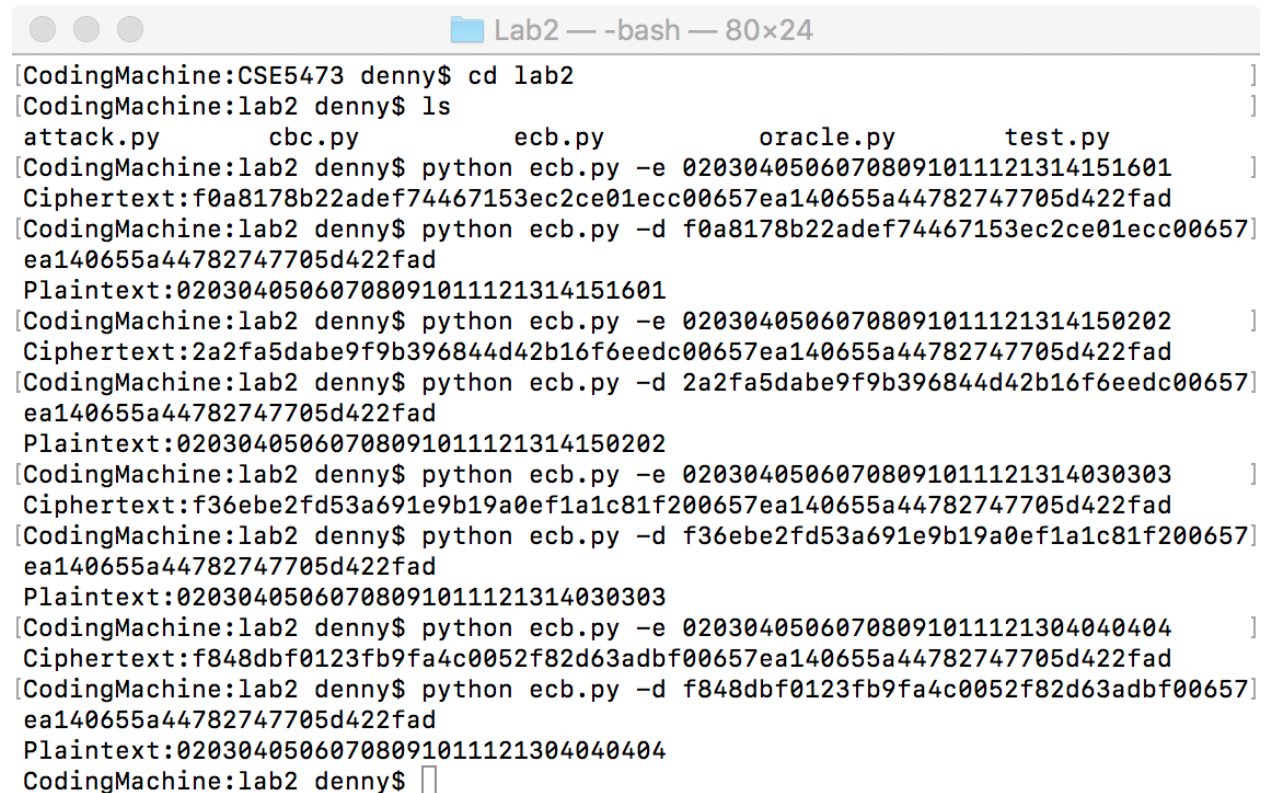


CSE5473_Lab2

Tianyuan Li

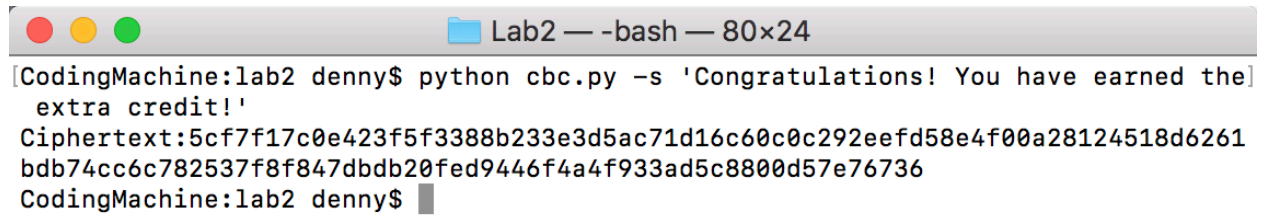
2.



```
[CodingMachine:CSE5473 denny$ cd lab2  
[CodingMachine:lab2 denny$ ls  
attack.py      cbc.py      ecb.py      oracle.py      test.py  
[CodingMachine:lab2 denny$ python ecb.py -e 02030405060708091011121314151601  
Ciphertext:f0a8178b22adef74467153ec2ce01ecc00657ea140655a44782747705d422fad  
[CodingMachine:lab2 denny$ python ecb.py -d f0a8178b22adef74467153ec2ce01ecc00657  
ea140655a44782747705d422fad  
Plaintext:02030405060708091011121314151601  
[CodingMachine:lab2 denny$ python ecb.py -e 02030405060708091011121314150202  
Ciphertext:2a2fa5dabe9f9b396844d42b16f6eedc00657ea140655a44782747705d422fad  
[CodingMachine:lab2 denny$ python ecb.py -d 2a2fa5dabe9f9b396844d42b16f6eedc00657  
ea140655a44782747705d422fad  
Plaintext:02030405060708091011121314150202  
[CodingMachine:lab2 denny$ python ecb.py -e 02030405060708091011121314030303  
Ciphertext:f36ebe2fd53a691e9b19a0ef1a1c81f200657ea140655a44782747705d422fad  
[CodingMachine:lab2 denny$ python ecb.py -d f36ebe2fd53a691e9b19a0ef1a1c81f200657  
ea140655a44782747705d422fad  
Plaintext:02030405060708091011121314030303  
[CodingMachine:lab2 denny$ python ecb.py -e 02030405060708091011121304040404  
Ciphertext:f848dbf0123fb9fa4c0052f82d63adbf00657ea140655a44782747705d422fad  
[CodingMachine:lab2 denny$ python ecb.py -d f848dbf0123fb9fa4c0052f82d63adbf00657  
ea140655a44782747705d422fad  
Plaintext:02030405060708091011121304040404  
CodingMachine:lab2 denny$
```

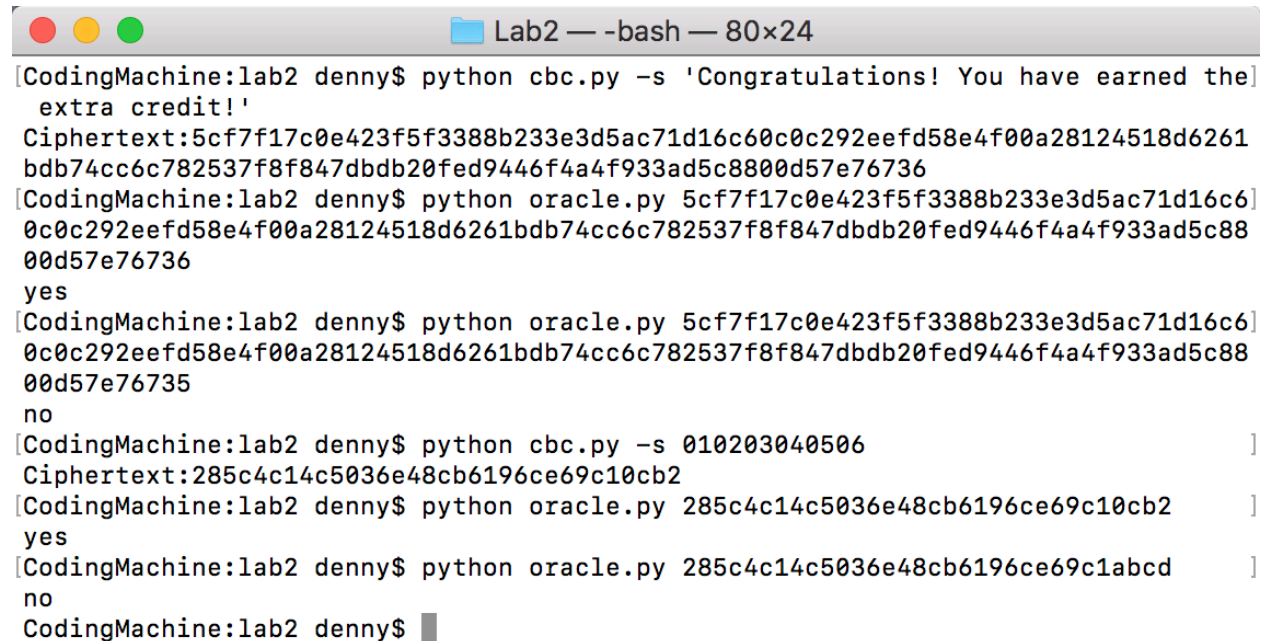
As shown in the figure above. After encrypting and then decrypting the hexadecimal strings with paddings, we get the original plaintext.

3.



```
[CodingMachine:lab2 denny$ python cbc.py -s 'Congratulations! You have earned the  
extra credit!'  
Ciphertext:5cf7f17c0e423f5f3388b233e3d5ac71d16c60c0c292eefd58e4f00a28124518d6261  
bdb74cc6c782537f8f847dbdb20fed9446f4a4f933ad5c8800d57e76736  
CodingMachine:lab2 denny$
```

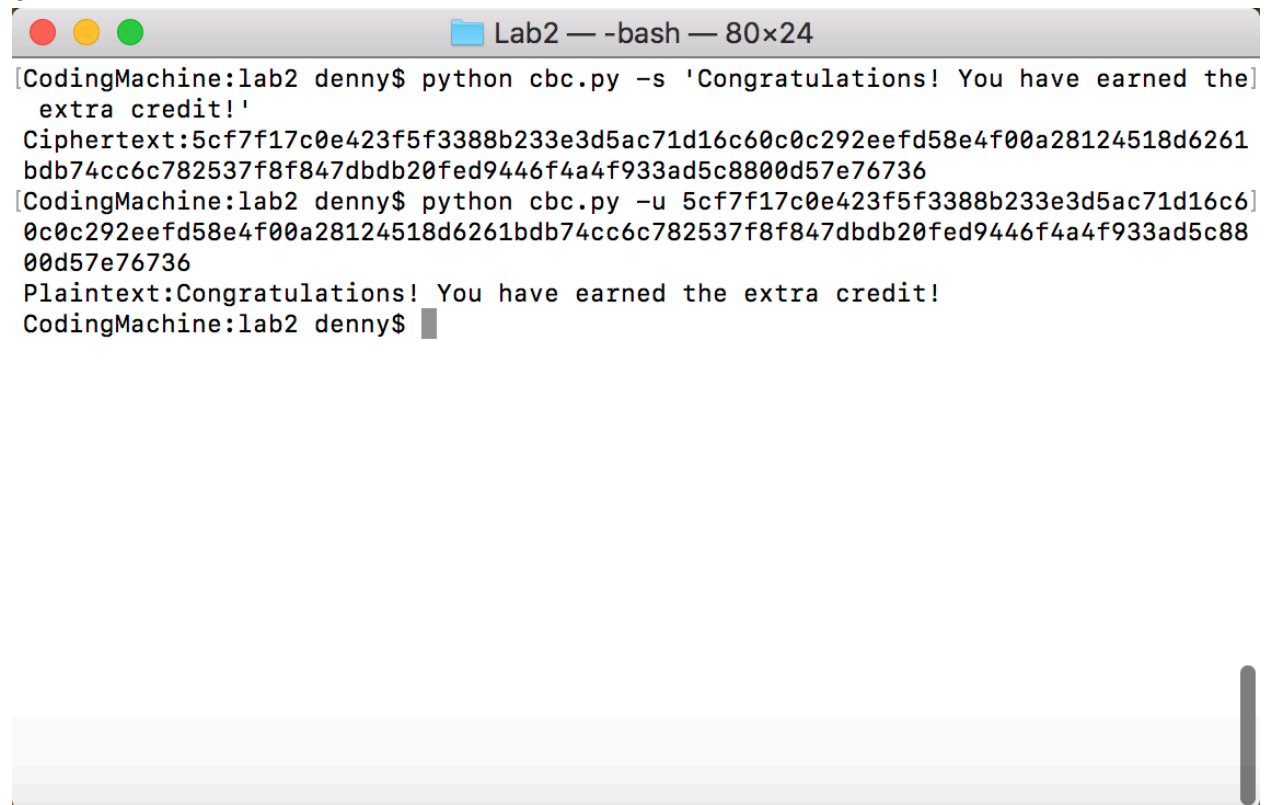
4.



```
Lab2 — -bash — 80x24
[CodingMachine:lab2 denny$ python cbc.py -s 'Congratulations! You have earned the
  extra credit!']
Ciphertext:5cf7f17c0e423f5f3388b233e3d5ac71d16c60c0c292eefd58e4f00a28124518d6261
bdb74cc6c782537f8f847dbdb20fed9446f4a4f933ad5c8800d57e76736
[CodingMachine:lab2 denny$ python oracle.py 5cf7f17c0e423f5f3388b233e3d5ac71d16c6]
0c0c292eefd58e4f00a28124518d6261bdb74cc6c782537f8f847dbdb20fed9446f4a4f933ad5c88
00d57e76736
yes
[CodingMachine:lab2 denny$ python oracle.py 5cf7f17c0e423f5f3388b233e3d5ac71d16c6]
0c0c292eefd58e4f00a28124518d6261bdb74cc6c782537f8f847dbdb20fed9446f4a4f933ad5c88
00d57e76735
no
[CodingMachine:lab2 denny$ python cbc.py -s 010203040506 ]
Ciphertext:285c4c14c5036e48cb6196ce69c10cb2
[CodingMachine:lab2 denny$ python oracle.py 285c4c14c5036e48cb6196ce69c10cb2 ]
yes
[CodingMachine:lab2 denny$ python oracle.py 285c4c14c5036e48cb6196ce69c1abcd ]
no
CodingMachine:lab2 denny$ █
```

As the figure shown, I tested the program by passing a correct padding ciphertext from encrypting test strings or hexadecimal strings and got “yes”. Then I modified a few bits of the correct padding ciphertext and got “no”.

6.



```
[CodingMachine:lab2 denny$ python cbc.py -s 'Congratulations! You have earned the
extra credit!'
Ciphertext:5cf7f17c0e423f5f3388b233e3d5ac71d16c60c0c292eefd58e4f00a28124518d6261
bdb74cc6c782537f8f847dbdb20fed9446f4a4f933ad5c8800d57e76736
[CodingMachine:lab2 denny$ python cbc.py -u 5cf7f17c0e423f5f3388b233e3d5ac71d16c6
0c0c292eefd58e4f00a28124518d6261bdb74cc6c782537f8f847dbdb20fed9446f4a4f933ad5c88
00d57e76736
Plaintext:Congratulations! You have earned the extra credit!
CodingMachine:lab2 denny$
```

As shown in the figure, I decrypted the entire cipher text and did not encounter difficulty.