

**Q1.**

```
from Crypto.Cipher import AES
from Crypto.Util import Counter
from Crypto import Random
import binascii

# AES supports multiple key sizes: 16 (AES128), 24 (AES192), or 32 (AES256).
key_bytes = 32

# Takes as input a 32-byte key and an arbitrary-length plaintext and returns a
# pair (iv, ciphertext). "iv" stands for initialization vector.
def encrypt(key, plaintext):
    assert len(key) == key_bytes

    # Choose a random, 16-byte IV.
    iv = Random.new().read(AES.block_size)

    # Convert the IV to a Python integer.
    iv_int = int(binascii.hexlify(iv), 16)

    # Create a new Counter object with IV = iv_int.
    ctr = Counter.new(AES.block_size * 8, initial_value=iv_int)

    # Create AES-CTR cipher.
    aes = AES.new(key, AES.MODE_CTR, counter=ctr)

    # Encrypt and return IV and ciphertext.
    ciphertext = aes.encrypt(plaintext)
    return (iv, ciphertext)

# Takes as input a 32-byte key, a 16-byte IV, and a ciphertext, and outputs the
# corresponding plaintext.
def decrypt(key, iv, ciphertext):
    assert len(key) == key_bytes

    # Initialize counter for decryption. iv should be the same as the output of
    # encrypt().
    iv_int = int(binascii.hexlify(iv), 16)
    ctr = Counter.new(AES.block_size * 8, initial_value=iv_int)

    # Create AES-CTR cipher.
    aes = AES.new(key, AES.MODE_CTR, counter=ctr)

    # Decrypt and return the plaintext.
```

```

    plaintext = aes.decrypt(ciphertext)
    return plaintext

# You need to define 32-byte key. Like
# Key = '12345678901234567890123456789012'
# Please refer this page on how to create key.
# Otherwise you will get this error.
key = '12345678901234567890123456789012'
(iv, ciphertext) = encrypt(key, 'How are you')
print(decrypt(key, iv, ciphertext))

```

The screenshot shows a Replit web interface with a Python script for AES encryption and decryption. The script is named 'main.py' and is located in the 'EnchantedGrimEystrain' repository. The script includes comments and code for key generation, encryption, and decryption. The output in the terminal shows the encrypted ciphertext 'b'\How are you\''.

```

18 # Convert the IV to a Python integer.
19 iv_int = int(binascii.hexlify(iv), 16)
20
21 # Create a new Counter object with IV = iv_int.
22 ctr = Counter.new(AES.block_size * 8, initial_value=iv_int)
23
24 # Create AES-CTR cipher.
25 aes = AES.new(key, AES.MODE_CTR, counter=ctr)
26
27 # Encrypt and return IV and ciphertext.
28 ciphertext = aes.encrypt(plaintext)
29 return (iv, ciphertext)
30
31 # Takes as input a 32-byte key, a 16-byte IV, and a ciphertext, and outputs the
32 # corresponding plaintext.
33 def decrypt(key, iv, ciphertext):
34     assert len(key) == key_bytes
35
36     # Initialize counter for decryption. iv should be the same as the output of
37     # encrypt().
38     iv_int = int(binascii.hexlify(iv), 16)
39     ctr = Counter.new(AES.block_size * 8, initial_value=iv_int)
40
41     # Create AES-CTR cipher.
42     aes = AES.new(key, AES.MODE_CTR, counter=ctr)
43
44     # Decrypt and return the plaintext.
45     plaintext = aes.decrypt(ciphertext)
46     return plaintext
47
48 # You need to define 32-byte key. Like
49 # Key = '12345678901234567890123456789012'
50 # Please refer this page on how to create key.
51 # Otherwise you will get this error.
52 key = '12345678901234567890123456789012'
53 (iv, ciphertext) = encrypt(key, 'How are you')
54 print(decrypt(key, iv, ciphertext))

```

The terminal output shows the encrypted ciphertext 'b'\How are you\''.

```

Python 3.6.1 (default, Dec 2015, 13:05:11)
[OS: 4.8.2] on linux
b'\How are you\'

```