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, ciphtertext). "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))
```