

```

# =====
'''crypto01da.py - cryptographic module - this file implements some basic
cryptographic functions - Divya Aradhya 5/20/2017 '''
# =====

import random, string

# -----
def RandomNulls(NumChar, LowChar, HighChar) :
    '''RandomNulls(NumChar, LowChar, HighChar) - returns NumChar
    random characters in the range LowChar to HighChar
    '''
    RetVal = ''
    LowNum = ord(LowChar)
    HighNum = ord(HighChar)
    for i in range(0, NumChar):
        RetVal = RetVal + chr(random.randint(LowNum,HighNum))

    return RetVal

# -----

def EncodeNullCipher(PlainText, lead, internal, trail):
    '''EncodeNullCipher(PlainText, lead, internal, trail) - encode the
    plaintext message with lead nulls in front, internal nulls
    between letters, and trail nulls at the end
    '''
    CipherText = ''
    CipherText = RandomNulls(lead, ' ', '~')
    for i in range(0, len(PlainText)-1):
        CipherText = CipherText + PlainText[i]
        CipherText = CipherText + RandomNulls(internal, ' ', '~')

    CipherText = CipherText + PlainText[i+1]
    CipherText = CipherText + RandomNulls(trail, ' ', '~')
    return CipherText

# -----

def DecodeNullCipher(CipherText, lead, internal, trail):
    '''DecodeNullcipher(CipherText, lead, internal, trail) -- decode
    the CipherText by removing the lead nulls and trail nulls, and
    keeping only every (internal+1)the character.
    This uses the slicing operator STRING[startIndex:endIndex:skip]
    '''

    return CipherText[lead: -trail: (internal+1)]

# -----

# =====
if __name__ == '__main__' :
    TestText = ['Lions', 'SaintLeo', 'Sunny Florida',
                string.digits, string.printable]

```

```
for Sample in TestText:
    Code = EncodeNullCipher(Sample,2,3,5)
    Decode = DecodeNullCipher(Code,2,3,5)
    print("Sample :" + Sample)
    print("Code :" + Code)
    print("Decode :" + Decode)
    if (Sample==Decode) :
        print('Round Trip OK\n')
    else :
        print('ERROR; Round Trip Mismatch\n')
# =====
```