33. Write a python program for Data encryption standard (DES) has been found vulnerable to very powerful attacks and therefore, the popularity of DES has been found slightly on the decline. DES is a block cipher and encrypts data in blocks of size of 64 bits each, which means 64 bits of plain text go as the input to DES, which produces 64 bits of ciphertext. The same algorithm and key are used for encryption and decryption, with minor differences. The key length is 56 bits. Implement in python programming.

Simple DES-like educational implementation

8-bit plaintext block, 10-bit key, 2-round Feistel for demonstration

Code:

```
def permute(bits, table):
    return [bits[i] for i in table]
def xor(bits1, bits2):
    return [b1 ^ b2 for b1, b2 in zip(bits1, bits2)]
def f_function(R, K):
    # simple F-function: expand R (4-bit -> 8-bit), XOR with K
    E = [0,1,3,2,3,2,0,1]
    R_exp = [R[i] for i in E]
    return xor(R_exp, K)
def encrypt_block(P, K1, K2):
    L, R = P[:4], P[4:]
    # Round 1
    L1 = R
    R1 = xor(L, f_function(R, K1))
    # Round 2
    L2 = R1
    R2 = xor(L1, f_function(R1, K2))
    return L2 + R2
def decrypt_block(C, K1, K2):
    # Decryption: apply keys in reverse
```

```
    L, R = C[:4], C[4:]
    # Round 1
    L1 = R
    R1 = xor(L, f_function(R, K2))
    # Round 2
    L2 = R1
    R2 = xor(L1, f_function(R1, K1))
    return L2 + R2
# Example
plaintext = [0,0,0,1, 0,0,1,1]  # 8 bits
key = [1,0,1,0,0,0,0,0,1,1]     # 10 bits
K1 = key[:8]
K2 = key[2:10]
cipher = encrypt_block(plaintext, K1, K2)
decrypted = decrypt_block(cipher, K1, K2)
print("Plaintext :", plaintext)
print("Ciphertext:", cipher)
print("Decrypted :", decrypted)
```

```
>>>
========== RESTART: C:/Users/Maria/OneDrive/Documents/ex33.py ==========
Plaintext : [0, 0, 0, 1, 0, 0, 1, 1]
Ciphertext: [1, 0, 0, 0, 0, 0, 1, 1]
Decrypted : [0, 0, 1, 1, 1, 0, 1, 0]
>>>
```