program for DES the first 24 bits of each subkey come from the same subset of 28 bits of the initial key and that the second 24 bits of each subkey come from a disjoint subset of 28 bits of the initial key.

# Simple DES Key Generation Demo

# Shows how first 24 bits come from C and next 24 bits from D

# --- PC-1: 64-bit key -> 56-bit key (drop parity bits) ---

PC1 = [

57,49,41,33,25,17,9,1,58,50,42,34,26,18,

10,2,59,51,43,35,27,19,11,3,60,52,44,36,

63,55,47,39,31,23,15,7,62,54,46,38,30,22,

14,6,61,53,45,37,29,21,13,5,28,20,12,4

]

# --- PC-2: selects 48 bits (24 from C and 24 from D) ---

PC2 = [

14,17,11,24,1,5,3,28,15,6,21,10,23,19,12,4,

26,8,16,7,27,20,13,2, # <-- from C (first 24)

41,52,31,37,47,55,30,40,51,45,33,48,

44,49,39,56,34,53,46,42,50,36,29,32 # <-- from D (next 24)

]

# --- Shift schedule for each round ---

SHIFT\_SCHEDULE = [1,1,2,2,2,2,2,2,1,2,2,2,2,2,2,1]

def left\_shift(bits, n):

"""Circular left shift."""

return bits[n:] + bits[:n]

def hex\_to\_bits(h):

"""Convert hex to list of bits."""

num = int(h, 16)

return [(num >> i) & 1 for i in reversed(range(len(h)\*4))]

def permute(bits, table):

"""Reorder bits according to table."""

return [bits[i-1] for i in table]

def generate\_subkeys(key\_hex):

"""Generate 16 DES subkeys and show left/right 24-bit parts."""

key\_bits = hex\_to\_bits(key\_hex)

key56 = permute(key\_bits, PC1) # apply PC1 (remove parity)

C, D = key56[:28], key56[28:] # split into halves

print("Initial 56-bit key split into:")

print("C0 =", ''.join(map(str, C)))

print("D0 =", ''.join(map(str, D)))

subkeys = []

for round\_num, shift in enumerate(SHIFT\_SCHEDULE, start=1):

C = left\_shift(C, shift)

D = left\_shift(D, shift)

CD = C + D

K = permute(CD, PC2)

subkeys.append(K)

# Show how the first 24 and next 24 come from C and D

print(f"\nRound {round\_num} Subkey (48 bits):")

print("From C (first 24 bits):", ''.join(map(str, K[:24])))

print("From D (next 24 bits): ", ''.join(map(str, K[24:])))

# --- Run demo ---

if \_\_name\_\_ == "\_\_main\_\_":

key\_hex = input("Enter 16-hex-character DES key: ").strip()

if len(key\_hex) != 16:

print("Key must be 16 hex characters (64 bits).")

else:

generate\_subkeys(key\_hex)

