1. Permutation choice-1 (PC-1) and final permutation (inverse of PC-1):

PC-1: Permutes a 64 bit key to remove parity bits, leaving 56 bits.

PC-2: (Final permutation choice): Reduces 56-bit combined key shares to 48 bit round keys

2. Compression Box: 64 → 48 bits

We'll simulate a design that selects 48-bits from a 64 bit input.

3. Expansion Box: 48 → 64 bits

we simulate a method to expand 48 bits to 64 by repeating some bits.

```
from typing import List

def permute(input_bits: str, table: List[int]
→ str: """permute input_bits according
to the table"""
return "".join(input_bits[i-1] for i in table)
```

#1. permutation choice-1 (PC-1): from 64-bit key to 56 bit key

PC1_TABLE = [

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

]

#2. compression Box: compress 64 to 48 bits (custom example)

compression_BOX = [

1, 3, 5, 7, 9, 11, 13, 15,
17, 19, 21, 23, 25, 27, 29, 31,
33, 35, 37, 39, 41, 43, 45, 47,
49, 51, 53, 55, 57, 59, 61, 63,
2, 4, 6, 8, 10, 12, 14, 16,
18, 20, 22, 24, 26, 28, 30, 32

]

#5. Expansion Box: Expand 48 bits to 64

(custom example)

Expansion_Box = [

1, 2, 3, 4, 5, 6, 7, 8,

5, 6, 7, 8, 9, 10, 11, 12,

9, 10, 11, 12, 13, 14, 15, 16,

13, 14, 15, 16, 17, 18, 19, 20,

17, 18, 19, 20, 21, 22, 23, 24,

21, 22, 23, 24, 25, 26, 27, 28,

25, 26, 27, 28, 29, 30, 31, 32,

1, 2, 3, 4, 5, 6, 7, 8

#Example 64 bits input (last binary string)

input_64bit = "0110110101110100010101010000111

1000110011011010100101111001

1010"

#1. Apply PC-1

key_56bit = permute (input_64bit, PC1_

TABLE)

print(f"56-bit key (pc-1): {key_56bit}")

#2. compress 64→ 48

compressed_48bits = permute(input_64bit, COMPRESSION_BOX)

print(f"compressed 48-bit output: {compressed_48bit}")

#3. Expand 48→64

Expanded_64bit = permute(compressed_48bit, EXPANSION_BOX)

print(f"Expanded 64bit output: {Expanded_64bit}").