By-Varun Singhal(MT20099)
Sachin Kotiyal(MT20093)

CSE 540 Network Security
Assignment 2
Project 0

Question: You are required to develop a program to encrypt (and similarly decrypt) a 64-bit plaintext using DES. Instead of using
an available library, I insist that you program any and every element of each of the 16 rounds of DES (and that means F-box, 32-
bit exchanges, generation of sub-key required in each round). Then, with at least TWO pairs of <64-bit plaintext, ciphertext>:

a. Verify that once the ciphertext is decrypted one gets the original plaintext,

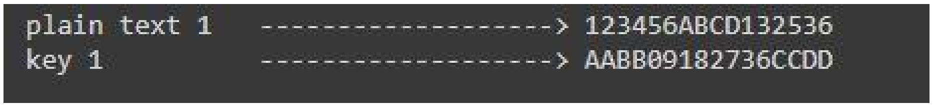
b. Verify that the output of the 1st encryption round is the same as the output of the 15th decryption round as illustrated below, and

c. Verify that the output of the 14th encryption round is the same as the output of the 2nd decryption round as illustrated below.

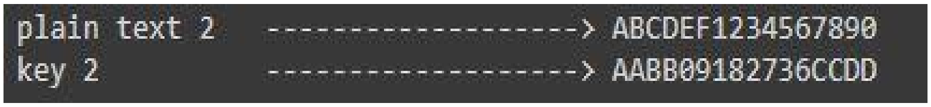
Solution Steps:

1. **Insert two 64 bit plaintext (p1 and p2) and two 64 bit keys(k1 & k2).**
2. **Generate 48 bits for every 16 rounds using those 64-bit keys.**
3. **Initialize initial and final permutation matrix.**
4. **Call encryption to get ciphertext for both plaintext(p1 & p2).**
5. **Then call decryption with reverse order of 16 round keys to get the plaintext.**
6. **Then verify text at each level of encryption and decryption as given in the question.**

Step1: Insert two plaintext(p1 & p2) and key of 64 bit

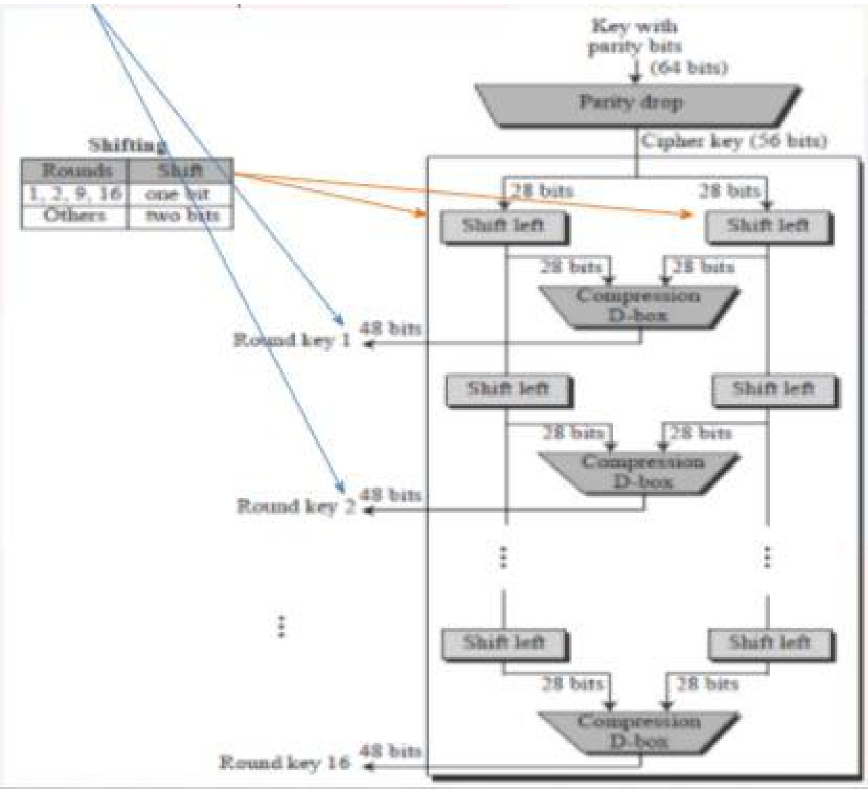


```
plain text 1    -------------------> 123456ABCD132536
key 1           -------------------> AABB09182736CCDD
```

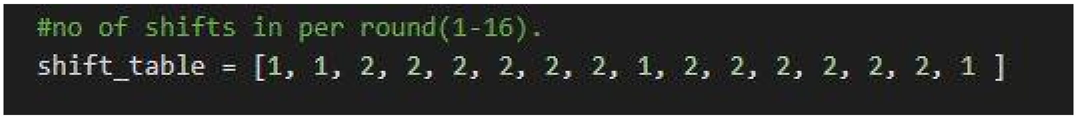


```
plain text 2    -------------------> ABCDEF1234567890
key 2           -------------------> AABB09182736CCDD
```

Step2: Generate 16 round keys of the 64-bit size of the main key.

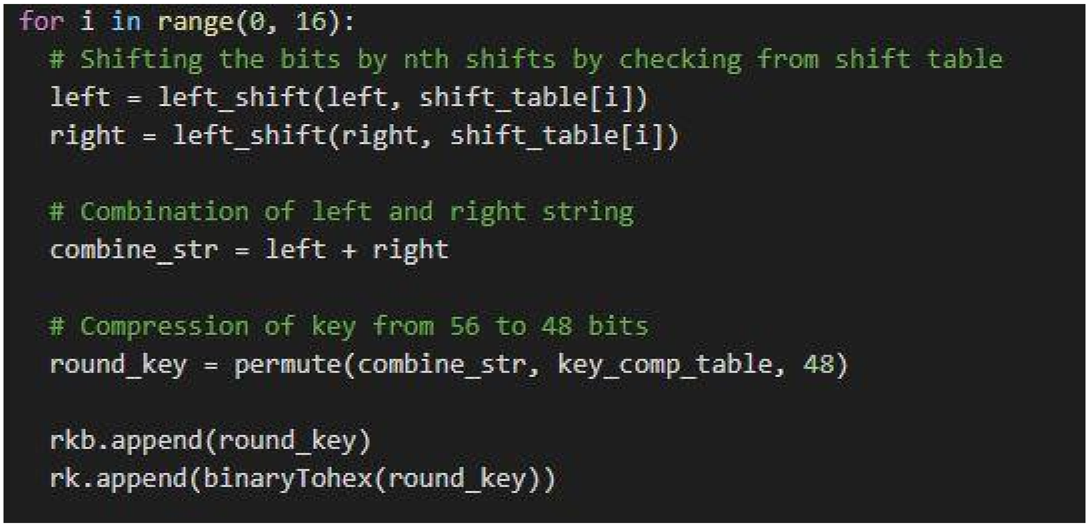


- **Parity drop (converted) into 56 bit key.**
- **Split plaintext into 2 parts i.e. left and right of 28 bits.**
- **We will left shift by how many bits, it depends on the round number given by the below table**



```
#no of shifts in per round(1-16).
shift_table = [1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1 ]
```

- **Using the help of compression D-box generates 48 bit keys in every round.**



```
for i in range(0, 16):
  # Shifting the bits by nth shifts by checking from shift table
  left = left_shift(left, shift_table[i])
  right = left_shift(right, shift_table[i])

  # Combination of left and right string
  combine_str = left + right

  # Compression of key from 56 to 48 bits
  round_key = permute(combine_str, key_comp_table, 48)

  rkb.append(round_key)
  rk.append(binaryTohex(round_key))
```

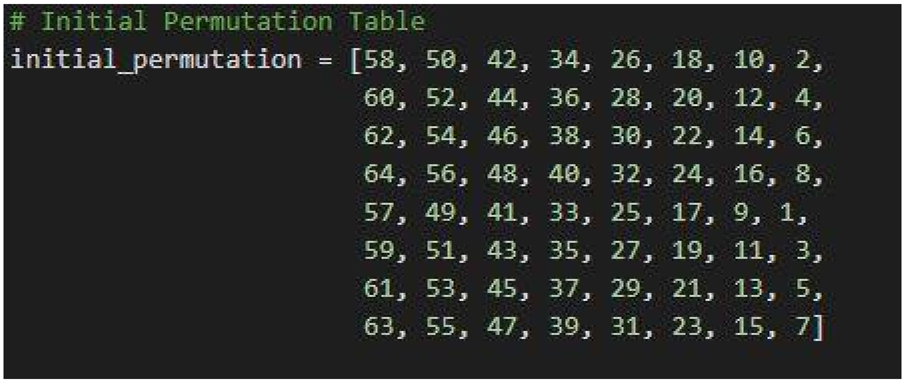
Step3: Initialize the initial and final permutation matrix

Based on the matrix given below, our permute function will make a string out of the given plaintext.

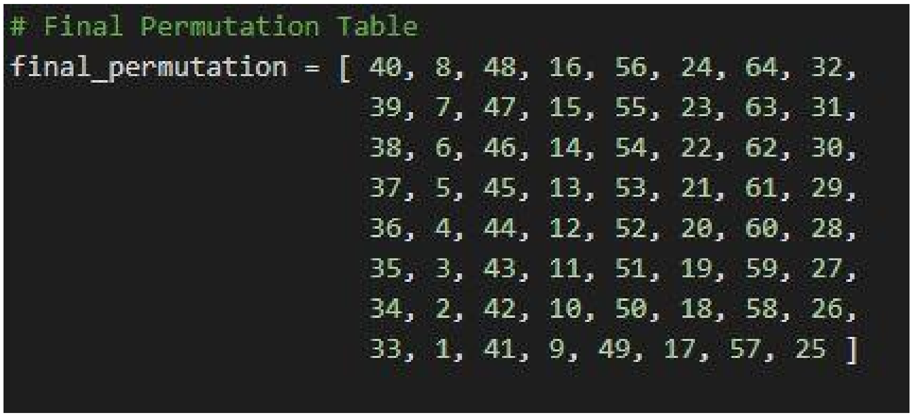
For example:-

The first value of the table suggests that the value at the 58th position will go to the 1st position.

The second value of the table suggests that the value at 50th position will go to 2nd position and so on



```
# Initial Permutation Table
initial_permutation = [58, 50, 42, 34, 26, 18, 10, 2,
                       60, 52, 44, 36, 28, 20, 12, 4,
                       62, 54, 46, 38, 30, 22, 14, 6,
                       64, 56, 48, 40, 32, 24, 16, 8,
                       57, 49, 41, 33, 25, 17, 9, 1,
                       59, 51, 43, 35, 27, 19, 11, 3,
                       61, 53, 45, 37, 29, 21, 13, 5,
                       63, 55, 47, 39, 31, 23, 15, 7]
```

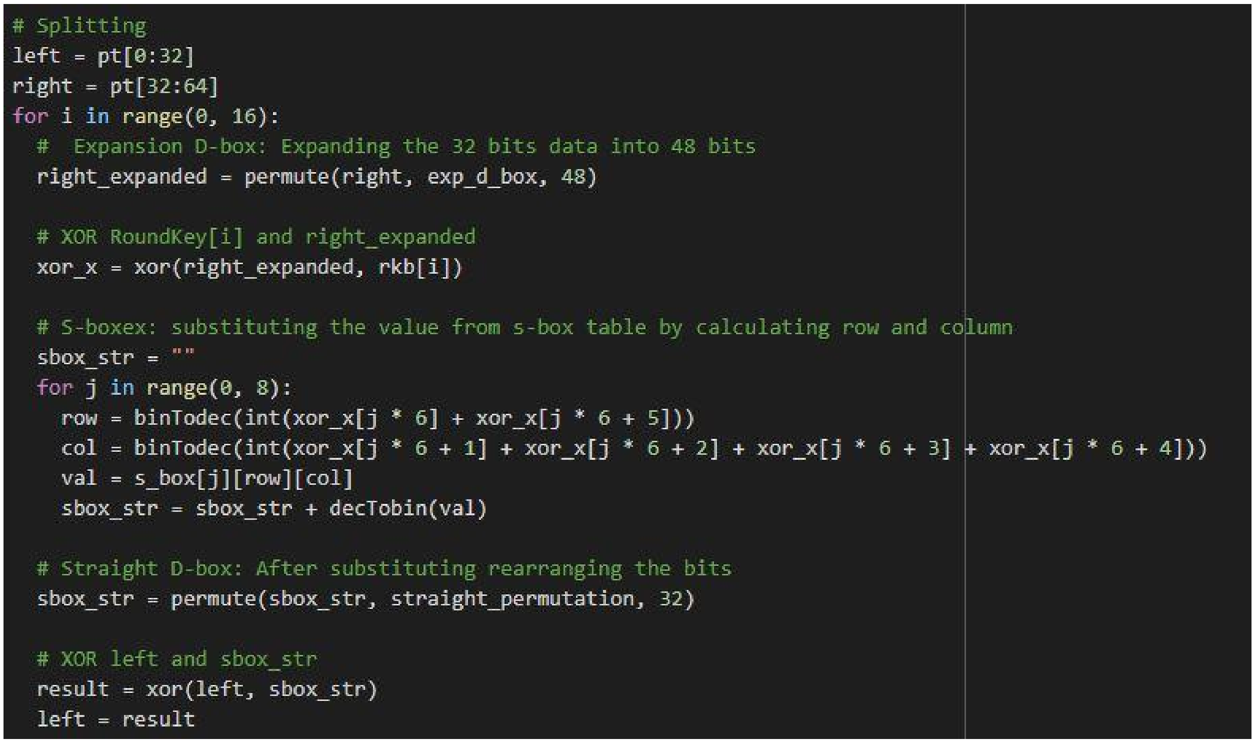


```
# Final Permutation Table
final_permutation = [ 40, 8, 48, 16, 56, 24, 64, 32,
                      39, 7, 47, 15, 55, 23, 63, 31,
                      38, 6, 46, 14, 54, 22, 62, 30,
                      37, 5, 45, 13, 53, 21, 61, 29,
                      36, 4, 44, 12, 52, 20, 60, 28,
                      35, 3, 43, 11, 51, 19, 59, 27,
                      34, 2, 42, 10, 50, 18, 58, 26,
                      33, 1, 41, 9, 49, 17, 57, 25 ]
```

Step4: DES encryption:

- After the initial permutation
- For each 16 round of algorithms**:**
    - Divide plaintext into two halves of 32 bit each.
    - Making a right 32 bit key to 48 bits using expansion P-box
    - We will XOR RoundKey[i] and right_expanded

  - ○ S-Box Substitution converts 48 bit to 32 bit after that we perform a straight P box.
  - ○ Perform xor of left and output of straight P box and swap left and right halves except last round.
- and then final permutation using the above matrix.



```
# Splitting
left = pt[0:32]
right = pt[32:64]
for i in range(0, 16):
  #  Expansion D-box: Expanding the 32 bits data into 48 bits
  right_expanded = permute(right, exp_d_box, 48)

  # XOR RoundKey[i] and right_expanded
  xor_x = xor(right_expanded, rkb[i])

  # S-boxex: substituting the value from s-box table by calculating row and column
  sbox_str = ""
  for j in range(0, 8):
    row = binTodec(int(xor_x[j * 6] + xor_x[j * 6 + 5]))
    col = binTodec(int(xor_x[j * 6 + 1] + xor_x[j * 6 + 2] + xor_x[j * 6 + 3] + xor_x[j * 6 + 4]))
    val = s_box[j][row][col]
    sbox_str = sbox_str + decTobin(val)

  # Straight D-box: After substituting rearranging the bits
  sbox_str = permute(sbox_str, straight_permutation, 32)

  # XOR left and sbox_str
  result = xor(left, sbox_str)
  left = result
```

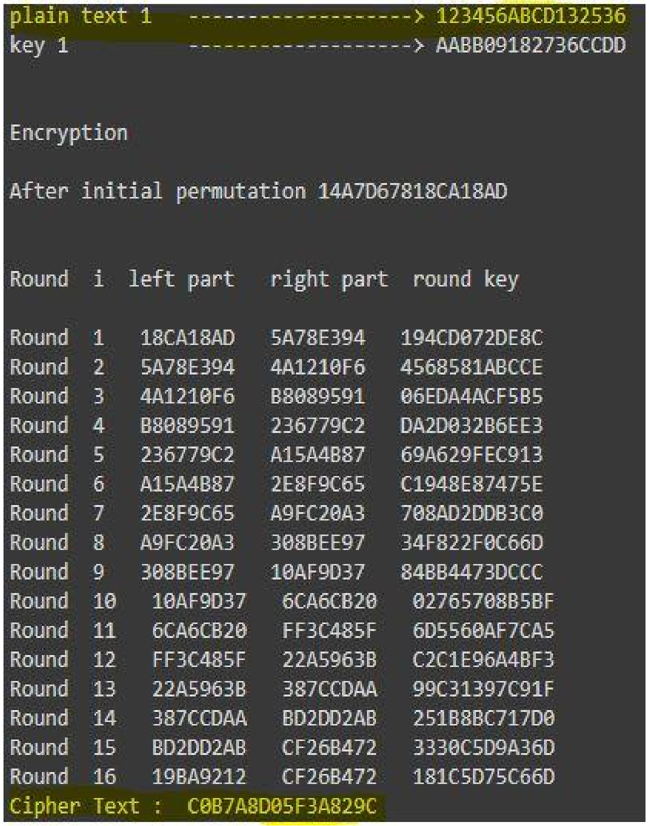
Step5: DES Decryption:

For description, code is written just to reverse the above steps in order to get the original output.

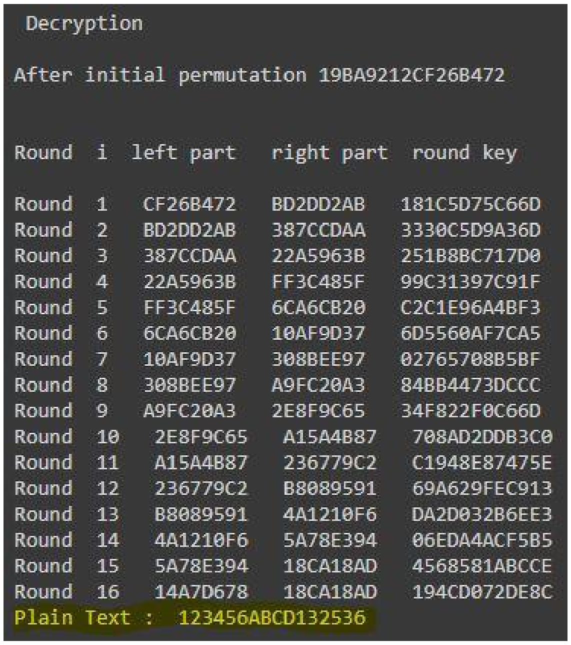
Step6:

Conclusion:

a) 1st pair of plaintext and ciphertext

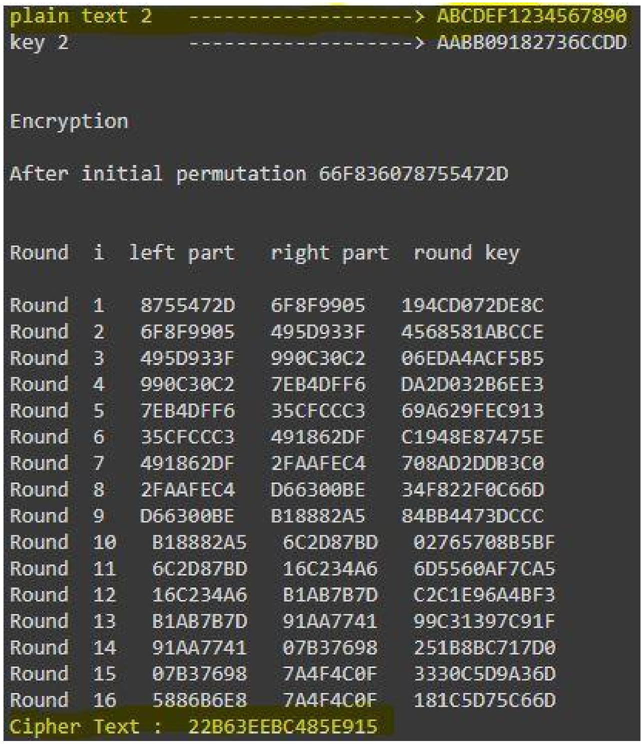


```
plain text 1   -------------------> 123456ABCD132536
key 1          -------------------> AABB09182736CCDD


Encryption

After initial permutation 14A7D67818CA18AD


Round  i  left part   right part  round key

Round  1   18CA18AD   5A78E394   194CD072DE8C
Round  2   5A78E394   4A1210F6   4568581ABCCE
Round  3   4A1210F6   B8089591   06EDA4ACF5B5
Round  4   B8089591   236779C2   DA2D032B6EE3
Round  5   236779C2   A15A4B87   69A629FEC913
Round  6   A15A4B87   2E8F9C65   C1948E87475E
Round  7   2E8F9C65   A9FC20A3   708AD2DDB3C0
Round  8   A9FC20A3   308BEE97   34F822F0C66D
Round  9   308BEE97   10AF9D37   84BB4473DCCC
Round  10   10AF9D37   6CA6CB20   02765708B5BF
Round  11   6CA6CB20   FF3C485F   6D5560AF7CA5
Round  12   FF3C485F   22A5963B   C2C1E96A4BF3
Round  13   22A5963B   387CCDAA   99C31397C91F
Round  14   387CCDAA   BD2DD2AB   251B8BC717D0
Round  15   BD2DD2AB   CF26B472   3330C5D9A36D
Round  16   19BA9212   CF26B472   181C5D75C66D
Cipher Text :  C0B7A8D05F3A829C
```

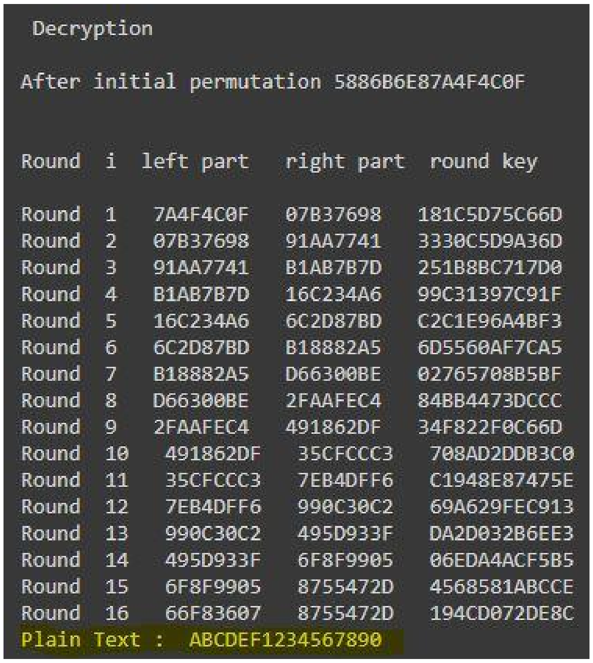


```
 Decryption

After initial permutation 19BA9212CF26B472


Round  i  left part   right part  round key

Round  1   CF26B472   BD2DD2AB   181C5D75C66D
Round  2   BD2DD2AB   387CCDAA   3330C5D9A36D
Round  3   387CCDAA   22A5963B   251B8BC717D0
Round  4   22A5963B   FF3C485F   99C31397C91F
Round  5   FF3C485F   6CA6CB20   C2C1E96A4BF3
Round  6   6CA6CB20   10AF9D37   6D5560AF7CA5
Round  7   10AF9D37   308BEE97   02765708B5BF
Round  8   308BEE97   A9FC20A3   84BB4473DCCC
Round  9   A9FC20A3   2E8F9C65   34F822F0C66D
Round  10   2E8F9C65   A15A4B87   708AD2DDB3C0
Round  11   A15A4B87   236779C2   C1948E87475E
Round  12   236779C2   B8089591   69A629FEC913
Round  13   B8089591   4A1210F6   DA2D032B6EE3
Round  14   4A1210F6   5A78E394   06EDA4ACF5B5
Round  15   5A78E394   18CA18AD   4568581ABCCE
Round  16   14A7D678   18CA18AD   194CD072DE8C
Plain Text :  123456ABCD132536
```

2nd pair of plaintext and ciphertext

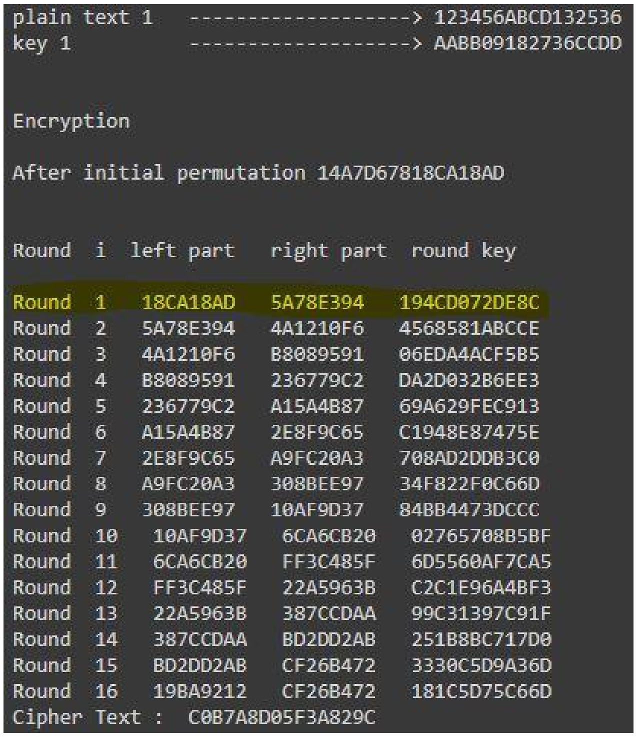


```
plain text 2   --------------------> ABCDEF1234567890
key 2          --------------------> AABB09182736CCDD


Encryption

After initial permutation 66F836078755472D


Round  i  left part   right part  round key

Round  1   8755472D   6F8F9905   194CD072DE8C
Round  2   6F8F9905   495D933F   4568581ABCCE
Round  3   495D933F   990C30C2   06EDA4ACF5B5
Round  4   990C30C2   7EB4DFF6   DA2D032B6EE3
Round  5   7EB4DFF6   35CFCCC3   69A629FEC913
Round  6   35CFCCC3   491862DF   C1948E87475E
Round  7   491862DF   2FAAFEC4   708AD2DDB3C0
Round  8   2FAAFEC4   D66300BE   34F822F0C66D
Round  9   D66300BE   B18882A5   84BB4473DCCC
Round  10   B18882A5   6C2D87BD   02765708B5BF
Round  11   6C2D87BD   16C234A6   6D5560AF7CA5
Round  12   16C234A6   B1AB7B7D   C2C1E96A4BF3
Round  13   B1AB7B7D   91AA7741   99C31397C91F
Round  14   91AA7741   07B37698   251B8BC717D0
Round  15   07B37698   7A4F4C0F   3330C5D9A36D
Round  16   5886B6E8   7A4F4C0F   181C5D75C66D
Cipher Text :  22B63EEBC485E915
```

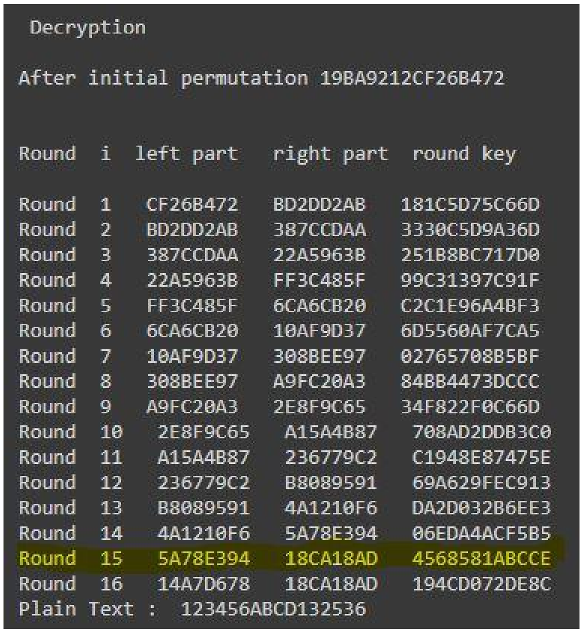


```
 Decryption

After initial permutation 5886B6E87A4F4C0F


Round  i  left part   right part  round key

Round  1   7A4F4C0F   07B37698   181C5D75C66D
Round  2   07B37698   91AA7741   3330C5D9A36D
Round  3   91AA7741   B1AB7B7D   251B8BC717D0
Round  4   B1AB7B7D   16C234A6   99C31397C91F
Round  5   16C234A6   6C2D87BD   C2C1E96A4BF3
Round  6   6C2D87BD   B18882A5   6D5560AF7CA5
Round  7   B18882A5   D66300BE   02765708B5BF
Round  8   D66300BE   2FAAFEC4   84BB4473DCCC
Round  9   2FAAFEC4   491862DF   34F822F0C66D
Round  10   491862DF   35CFCCC3   708AD2DDB3C0
Round  11   35CFCCC3   7EB4DFF6   C1948E87475E
Round  12   7EB4DFF6   990C30C2   69A629FEC913
Round  13   990C30C2   495D933F   DA2D032B6EE3
Round  14   495D933F   6F8F9905   06EDA4ACF5B5
Round  15   6F8F9905   8755472D   4568581ABCCE
Round  16   66F83607   8755472D   194CD072DE8C
Plain Text :  ABCDEF1234567890
```

b) We conclude that the output of the 1st encryption round is the same as the output of the 15th decryption round as shown below:



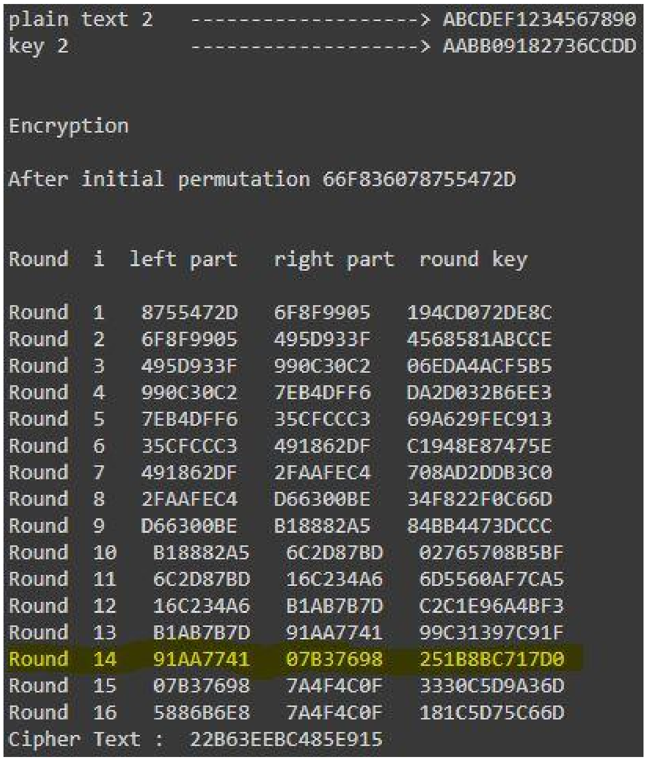
```
plain text 1   -------------------> 123456ABCD132536
key 1          -------------------> AABB09182736CCDD


Encryption

After initial permutation 14A7D67818CA18AD


Round  i  left part   right part  round key

Round  1   18CA18AD    5A78E394    194CD072DE8C
Round  2   5A78E394    4A1210F6    4568581ABCCE
Round  3   4A1210F6    B8089591    06EDA4ACF5B5
Round  4   B8089591    236779C2    DA2D032B6EE3
Round  5   236779C2    A15A4B87    69A629FEC913
Round  6   A15A4B87    2E8F9C65    C1948E87475E
Round  7   2E8F9C65    A9FC20A3    708AD2DDB3C0
Round  8   A9FC20A3    308BEE97    34F822F0C66D
Round  9   308BEE97    10AF9D37    84BB4473DCCC
Round  10   10AF9D37    6CA6CB20    02765708B5BF
Round  11   6CA6CB20    FF3C485F    6D5560AF7CA5
Round  12   FF3C485F    22A5963B    C2C1E96A4BF3
Round  13   22A5963B    387CCDAA    99C31397C91F
Round  14   387CCDAA    BD2DD2AB    251B8BC717D0
Round  15   BD2DD2AB    CF26B472    3330C5D9A36D
Round  16   19BA9212    CF26B472    181C5D75C66D
Cipher Text :  C0B7A8D05F3A829C
```



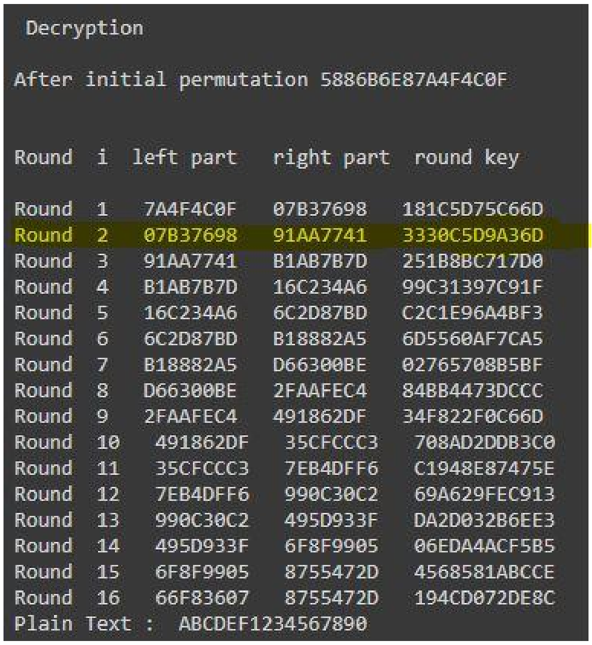
```
 Decryption

After initial permutation 19BA9212CF26B472


Round  i  left part   right part  round key

Round  1   CF26B472    BD2DD2AB    181C5D75C66D
Round  2   BD2DD2AB    387CCDAA    3330C5D9A36D
Round  3   387CCDAA    22A5963B    251B8BC717D0
Round  4   22A5963B    FF3C485F    99C31397C91F
Round  5   FF3C485F    6CA6CB20    C2C1E96A4BF3
Round  6   6CA6CB20    10AF9D37    6D5560AF7CA5
Round  7   10AF9D37    308BEE97    02765708B5BF
Round  8   308BEE97    A9FC20A3    84BB4473DCCC
Round  9   A9FC20A3    2E8F9C65    34F822F0C66D
Round  10   2E8F9C65    A15A4B87    708AD2DDB3C0
Round  11   A15A4B87    236779C2    C1948E87475E
Round  12   236779C2    B8089591    69A629FEC913
Round  13   B8089591    4A1210F6    DA2D032B6EE3
Round  14   4A1210F6    5A78E394    06EDA4ACF5B5
Round  15   5A78E394    18CA18AD    4568581ABCCE
Round  16   14A7D678    18CA18AD    194CD072DE8C
Plain Text :  123456ABCD132536
```

c)

We conclude that the output of the 14th encryption round is the same as the output of the 2nd decryption round as shown below:



```
plain text 2   -------------------> ABCDEF1234567890
key 2          -------------------> AABB09182736CCDD


Encryption

After initial permutation 66F836078755472D


Round  i  left part   right part  round key

Round  1   8755472D   6F8F9905   194CD072DE8C
Round  2   6F8F9905   495D933F   4568581ABCCE
Round  3   495D933F   990C30C2   06EDA4ACF5B5
Round  4   990C30C2   7EB4DFF6   DA2D032B6EE3
Round  5   7EB4DFF6   35CFCCC3   69A629FEC913
Round  6   35CFCCC3   491862DF   C1948E87475E
Round  7   491862DF   2FAAFEC4   708AD2DDB3C0
Round  8   2FAAFEC4   D66300BE   34F822F0C66D
Round  9   D66300BE   B18882A5   84BB4473DCCC
Round  10   B18882A5   6C2D87BD   02765708B5BF
Round  11   6C2D87BD   16C234A6   6D5560AF7CA5
Round  12   16C234A6   B1AB7B7D   C2C1E96A4BF3
Round  13   B1AB7B7D   91AA7741   99C31397C91F
Round  14   91AA7741   07B37698   251B8BC717D0
Round  15   07B37698   7A4F4C0F   3330C5D9A36D
Round  16   5886B6E8   7A4F4C0F   181C5D75C66D
Cipher Text :  22B63EEBC485E915
```



```
 Decryption

After initial permutation 5886B6E87A4F4C0F


Round  i  left part   right part  round key

Round  1   7A4F4C0F   07B37698   181C5D75C66D
Round  2   07B37698   91AA7741   3330C5D9A36D
Round  3   91AA7741   B1AB7B7D   251B8BC717D0
Round  4   B1AB7B7D   16C234A6   99C31397C91F
Round  5   16C234A6   6C2D87BD   C2C1E96A4BF3
Round  6   6C2D87BD   B18882A5   6D5560AF7CA5
Round  7   B18882A5   D66300BE   02765708B5BF
Round  8   D66300BE   2FAAFEC4   84BB4473DCCC
Round  9   2FAAFEC4   491862DF   34F822F0C66D
Round  10   491862DF   35CFCCC3   708AD2DDB3C0
Round  11   35CFCCC3   7EB4DFF6   C1948E87475E
Round  12   7EB4DFF6   990C30C2   69A629FEC913
Round  13   990C30C2   495D933F   DA2D032B6EE3
Round  14   495D933F   6F8F9905   06EDA4ACF5B5
Round  15   6F8F9905   8755472D   4568581ABCCE
Round  16   66F83607   8755472D   194CD072DE8C
Plain Text :  ABCDEF1234567890
```

Thank You…….!