Question 1:

Show the Cipher text after one round of DES for the following plain-text and key in

hexadecimal numbers. Compare part a versus part b to verify the avalanche effect in DES.

a) Plain-text: 0 1 2 3 4 5 6 7 8 9 A B C D E F, Key: 0 1 2 3 4 5 6 7 8 9 A B C D E E

b) Plain-text: 0 1 2 3 4 5 6 7 8 9 A B C D E E, Key: 0 1 2 3 4 5 6 7 8 9 A B C D E F

solution:

a) Key creation:

Key: 0 1 2 3 4 5 6 7 8 9 A B C D E E

Convert to binary:

0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1110

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 61 | 62 | 63 | 64 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 1 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Apply Permutation choice 1:

Key +:

1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0

Divide the into left and right:

Key left: 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0

Key right: 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0

Then apply left shift table:

Key left\_1: 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1

Key right\_1: 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1

Key left\_2: 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1

Key right\_2: 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0

Key left\_3: 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1

Key right\_3: 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0

Key left\_4: 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0

Key right\_4: 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0

Key left\_5: 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0

Key right\_5: 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0

Key left\_6: 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1

Key right\_6: 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1

Key left\_7: 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0

Key right\_7: 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0

Key left\_8: 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1

Key right\_8: 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1

Key left\_9: 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0

Key right\_9: 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0

Key left\_10: 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1

Key right\_10: 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1

Key left\_11: 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1

Key right\_11: 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1

Key left\_12: 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1

Key right\_12: 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0

Key left\_13: 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1

Key right\_13: 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0

Key left\_14: 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0

Key right\_14: 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0

Key left\_15: 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0

Key right\_15: 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0

Key left\_16: 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0

Key right\_16: 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0

Concatenate left with right then apply permutation choice 2:

Key\_1: 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1

Key\_2: 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0

Key\_3: 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0

Key\_4: 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0

Key\_5: 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0

Key\_6: 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1

Key \_7: 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0

Key\_8: 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1

Key\_9: 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0

Key\_10: 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1

Key\_11: 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1

Key\_12: 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0

Key\_13: 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0

Key\_14: 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0

Key\_15: 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0

Key\_16: 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0

Then apply permutation choice 2 to get 16 sub keys with 48 bits:

Key\_1: 000010110000001001100111100110110100100110100101

Key\_2: 011010011010011001011001001001010110101000100110

Key\_3: 010001011101010010001010101101000010100011010010

Key\_4: 011100101000100111010010101001011000001001010111

Key\_5: 001111001110100000000011000101111010011011000010

Key\_6: 001000110010010100011110001111001000010101000101

Key\_7: 011011000000010010010101000010101110010011000110

Key\_8: 010101111000100000111000011011001110010110000001

Key\_9: 110000001100100111101001001001101011100000111001

Key\_10:100100011110001100000111011000110001110101110010

Key\_11: 001000010001111110000011000011011000100100111010

Key\_12: 011100010011000011100101010001010101110001010100

Key\_13: 100100011100010011010000010010011000000011111100

Key\_14: 010101000100001110110110100000011101110010001101

Key\_15: 101101101001000100000101000010100001011010110101

Key\_16: 110010100011110100000011101110000111000000110010

Plain text encryption:

Convert to binary:

Plain text:

0000000100100011010001010110011110001001101010111100110111101111

Apply Initial Permutation:

1100110000000000110011001111111111110000101010101111000010101010

Divide to left and right:

L\_0: 11001100000000001100110011111111

R\_0: 11110000101010101111000010101010

Apply function:

L\_1=R\_0

R\_1=L\_0+F(R\_0,K\_1)

L\_1=11110000101010101111000010101010

R\_1=L\_0+F(R\_0,K\_1)

First calculate F(R\_0,K\_1)

Key 1 is 48 bit, and R\_0 is 32 bit so apply expansion table to R\_0:

R\_O after expansion: 011110100001010101010101011110100001010101010101

Key\_1: 000010110000001001100111100110110100100110100101

==============================================================

011100010001011100110010111000010101110011110000

==============================================================

then calculate L\_0+F(R\_0,K\_1) but the result of the function is 48 bit and L\_0 is 32

so apply s-box to the result of the function.

first divide it into 8 blocks each with 6 bits:

Apply s-box 1: 011100

take the first and last bit : 00 convert to decimal =0 indicate row

take the in between 4 bit : 1110 convert to decimal =14 indicate column

The intersection between row 0, column 14 in S-box\_1 is 0 :0000

Apply s-box 2: 010001

take the first and last bit : 01 convert to decimal =1 indicate row

take the in between 4 bit : 1000 convert to decimal =8 indicate column

The intersection between row 1, column 8 in S-box\_1 is 12 :1100

Apply s-box 3: 011100

take the first and last bit : 00 convert to decimal =0 indicate row

take the in between 4 bit : 1110 convert to decimal =14 indicate column

The intersection between row 0, column 14 in S-box\_1 is 2 :0010

Apply s-box 4: 110010

take the first and last bit : 10 convert to decimal =2 indicate row

take the in between 4 bit : 1001 convert to decimal =9 indicate column

The intersection between row 2, column 9 in S-box\_1 is 1 :0001

Apply s-box 5: 111000

take the first and last bit : 10 convert to decimal =2 indicate row

take the in between 4 bit : 1100 convert to decimal =12 indicate column

The intersection between row 2, column 12 in S-box\_1 is 6 :0110

Apply s-box 6: 010101

take the first and last bit : 01 convert to decimal =1 indicate row

take the in between 4 bit : 1010 convert to decimal =10 indicate column

The intersection between row 1, column 10 in S-box\_1 is 13 :1101

Apply s-box 7: 110011

take the first and last bit : 11 convert to decimal =3 indicate row

take the in between 4 bit : 1001 convert to decimal =9 indicate column

The intersection between row 3, column 9 in S-box\_1 is 5 :0101

Apply s-box 8: 110000

take the first and last bit : 10 convert to decimal =2 indicate row

take the in between 4 bit : 1000 convert to decimal =8 indicate column

The intersection between row 2, column 8 in S-box\_1 is 0 :0000

The result is after applying S-boxes: 00001100001000010110110101010000

Then apply primitive permutation Function: 10010010000111000010000010011100

Then apply L\_0+F=

L\_0 = 11001100000000001100110011111111

F = 10010010000111000010000010011100

====================================

01011110000111001110110001100011

====================================================================

b)