Section 4.2.1.(a): What are L0, R0?

L0=100110, R0=101101

Section 4.2.2.(b): Why does reversing order of the keys cause a reversal of the encryption? (Hint: given a value A, what is A XOR A? Given a value B, what is B XOR 0?)

A XOR A only produce 0, whatever the input A is. B XOR 0 has two situations: If B = 0, the output is 0, if B = 1, the output is 1. If the binary number is same, the output is same, if your binary number is different the output is different, so you we reverse the order of the keys, the binary number is also changed and it will also reverse the encryption result.

Section 4.2.3.(b).i

Output: 111

Section 4.2.3.(b).ii

Output: 111

Section 4.2.3.(c).iv (figure out the key iterations)

K1 = 11100011

K3 = 01111110

Section 4.2.4 Show your work!

Round 1 (*i* = 0)

1. *L*0= 100110 and *R*0= 101101; *K*1= 10110000.
2. *E*(*R*0) = 11011110.
3. *E*(*R*0)   *K*1 = 11011110  10110000 = 01101110.
4. S1(0110) = 011, S2(1110) = 000 🡪*f* (*R*0,*K*1) = 011000
5. *f* (*R*0,*K*1)  *L*0= 011000  100110= 111110.
6. Could we have had another DES-type algorithm if we had just picked different permutations for the initial and round-by-round keys?

It could have had another DES-type algorithm, such as 3DES Algorithms (Data Encryption Standard and Triple DES, it is still in use where a DES is run three times in sequence using distinct keys (which triples the size of the secret key to be shared). However, using 3DES mitigates this issue at the cost of increasing execution time.

1. Is there anything special about the s-boxes? Could they differ?

I think the special about the s-boxes is that Each S-box replaces a 6-bit input with a 4-bit output. Given a 6-bit input, the 4-bit output is found by selecting the row using the outer two bits, and the column using the inner four bits. It should be different as a standard, however, they could differ by modifying the S-Box.

1. Why might an encryption algorithm hard-code the permutations and s-boxes?

Because the hard-code the permutation and S-boxes can keep the encryption algorithm easier. it gives us a standard to search the binary.

1. Given what we know about decryption being the reversal of encryption, what would a really bad set of keys be?

Encryption reversal process is referred to as decryption. An encryption process has a corresponding decryption process, which is used to reverse the encrypted data (ciphertext) back to its original content (plaintext). I guess the bad set of keys could be all 0 or all 1.