

Question 1

For any P and C $E_K(P) = C$ and $E_{\bar{K}}(\bar{P}) = \bar{C}$. This can be proven by looking at each step of the DES algorithm individually.

- Permutations - All permutations in DES just rearrange the bits of their input. They do not change the bits in any way, just their order. Because the bits themselves aren't changed if $P(X) = Y$ then $P(\bar{X}) = \bar{Y}$ for any permutation.
- IP - The initial permutation just rearranges the bits of the input. The output of the IP for $E_{\bar{K}}(\bar{P})$ is therefore the bitwise complement of the output for the IP of $E_K(P)$.
- Key Schedule - The key schedule just rearranges the bits of the input values (and spreads them across 16 smaller words. The only operation it does on the input to the key schedule is a permutation (which like all other permutations in DES will only change the order of the bits) and shifts. This means ALL the generated keys from $S(\bar{K})$ will be the complement of the corresponding keys generated by $S(K)$.
- Feistel Function - For the first round the Feistel function's input is one subkey and a half of the plaintext. With an inverted plaintext and key the input for F in $E_{\bar{K}}$ case is \bar{P}' and \bar{K}' (where P' and K' represent the subkey and plaintext block). We can show that with both inputs inverted the output will be the same as it is without the inversion.
 - XOR - The first operation F does is XOR P' and K' . Because $X \oplus Y = \bar{X} \oplus \bar{Y}$ (any differences between X and Y will exist in both their original form and in their inverses) after this operation the F function continues using the exact same data as it would have with P' and K' in the $E_K(P)$ case.
 - SBoxes - The S-Box stage works exactly the same (and produces the same output) as it does in the $E_K(P)$ case because the input is the same (as a result of the XOR operation).
 - P - The permutation in F just rearranges the bits, but even if it did something more complicated that would still be ok because the input to the permutation is exactly the same as it is in the $E_K(P)$ case.
- Rounds - We know that the input to F for the first round is \bar{P}' and \bar{K}' and the output is the same as it is in the $E_K(P)$ case. We can show that the input to each round of F is \bar{P}' and \bar{K}' and the output is therefore the same as the corresponding output in $E_K(P)$.

The output of F is XORed with the other half of the plaintext (\bar{L}_0). This means we're XORing the output of F (which is the same as in the $E_K(P)$ case with the inverse of L_0). This means R_0 (the output of the XOR operation) is the inverse of what it is in the $E_K(P)$ case (because L_0 is inverted). \bar{R}_0 is unchanged in the first round and continues onto the next round as \bar{L}_1 .

This all means the output of each round (L_n and R_n) is always the inverse of what it would have been in the $E_K(P)$ case.

$$\begin{array}{ll}
L_n = R_{n-1} & \text{With } R_{n-1} \text{ inverted, so will } L_n \\
R_n = L_{n-1} \oplus F(R_{n-1}, K_n) & \text{With } R_{n-1}, L_{n-1} \text{ and } K_n \text{ inverted } R_n \text{ will also be}
\end{array}$$

- FP - After all 16 rounds we end up with L_{16}^{-} and R_{16}^{-} which are then combined and sent through the final permutation (which is the inverse of IP). As with other permutations, this just rearranges the bits so if the input is inverted (which it is) the output will be too. This is the end of the cipher.

Question 2

When $K = 111 \dots 111$ then $E_K(E_K(X))$ is the identity function. This is because for DES decryption is encryption with a reversed key schedule. When $K = 111 \dots 111$ $K_{0..15}$ all equal $111 \dots 111$, so the reversed key schedule is exactly the same as the original key schedule. E_K and D_K are therefore the same function when $K = 111 \dots 111$.

Another key that has this property is the key containing all zeros.

By showing that the decryption function is the inverse of the encryption function we can show that E_K is its own inverse when $K = 111 \dots 111$.

After the last round of encryption we have $R_n L_n$ (because the two sides aren't reversed after the last round of DES). The first round of decryption turns this into:

$$\begin{array}{ll}
L'_0 = R_n & \\
R'_0 = L_n & \\
L'_1 = R'_0 = L_n & \\
R'_1 = L'_0 \oplus F(R'_0, K_n) = R_n \oplus F(L_n, K_n) & \text{Note: } K_n, \text{ not } K_1
\end{array}$$

Using what we know about L_n and R_n we can make the following substitutions.

$$\begin{array}{l}
L'_1 = R_{n-1} \\
R'_1 = (L_{n-1} \oplus F(R_{n-1}, K_n)) \oplus F(L_n, K_n) \\
= L_{n-1} \oplus F(R_{n-1}, K_n) \oplus F(R_{n-1}, K_n) \\
= L_{n-1} \oplus 0 = L_{n-1}
\end{array}$$

As we continue through each subsequent decryption round $R_{n-1} L_{n-1}$ will be turned into $R_{n-2} L_{n-2}$, etc until we finally reach $R_0 L_0$. When the lack of a flip in the last round is taken into account we see that this results in the original plaintext.

This shows that D_K is the inverse of E_K . If E_K and D_K are actually the same function (as in the case when $K = 0$ or $K = \bar{0}$) then $E_K(E_K(X))$ is the identity function.