Ex 3.6 Let G be a pseudorandom generator where $|G(s)| \geq 2 \cdot |s|$.

(a) Define $G'(s) \overset{def}{=} G(s0^{|s|})$. Is G' necessarily a pseudorandom generator?

Answer: No, the function G' is not necessarily a pseudorandom generator.

Proof. Let G'' be any pseudorandom generator that has expansion factor $l(n) = 4 \cdot n$. If we define $H(s) = H(s_1, s_2) = G''(s_2)$, which $|s| = 2n, |s_1| = |s_2| = n$. We can show that H is a pseudorandom generator by reduction. Otherwise, assume that there's a distinguisher D which can distinguish $H(s)$ from a truly random string $w(|w| = 4n)$ with probability non-negligible.

$$|Pr[D(w) = 1] - Pr[D(H(s)) = 1]| > negl(n),$$

We can then construct a PPT distinguisher D' for psedorandom generagor G'' as follows: When receives input string w, which $|w| = 4n$, D' delivers w to D and outputs whatever D outputs. If w is generated by pseudorandom function G' with some random seed s' at length n, then the probability for D to succeed is the same as D' since the distribution of H and G'' over $\{0,1\}^{4n}$ is the same. So we have:

$$Pr[D'(G''(s')) = 1] = Pr[D(H(\{0,1\}^n|s')) = 1]$$

If w is chosen uniformly at random, then

$$Pr[D'(w) = 1] = Pr[D(w) = 1] = \frac{1}{2}$$

Thus $|Pr[D'(w) = 1] - Pr[D'(G''(s')) = 1] = |Pr[D(w) = 1] - Pr[D(H(s)) = 1]| > negl(n)$. Hence we can conclude that $H(s)$ is a pseudorandom generator. What if we replace G in our question with H? Now, $G'(s) = G(s0^{|s|} = H(s0^{|s|}) = G''(0^{|s|})$, which is obviously not pseudorandom generator. $\square$

(b) Define $G'(s) \overset{def}{=} G(s_1 \ldots s_{n/2})$, where $s = s_1 \ldots s_n$. Is G' necessarily a pseudorandom generagor?

Answer: Yes, the function G' is necessarily a pseudorandom generator. The expansion condition is preserved since $|G'(s)| = |G(s_1, \cdots, s_{n/2}| > 2 \cdot |n/2| = n$. And its pseudorandomness condition is satisfied too, which can be shown easily by reduction.

Ex 3.10 Let G be a pseudorandom generator and define $G'(s)$ to be the output of G truncated to n bits (where s is of length n). Prove that the function $F_k(x) = G'(k) \oplus x$ is not pseudorandom.

Proof. We can make a polynomial-time distinguisher D act as follows:

1. D outputs a random plaintext $m \in \{0,1\}^n$.
2. A random key k is fixed and used by Gen. The ciphertext $c \leftarrow Gen_k(m)$ is generated and given to D.
3. D calucates: $r = m\ XOR\ c$ and records r. Then D checks that if this r ever appeared previously. If it does, D outputs 0, which means it is pseudorandom. Otherwise, D outputs 1, which means it is random.

This will work since $G'(k)$ is fixed and everytime a ciphertext is generated with $F_k(x)$, it will use the same $G'(k)$. Thus an XOR of the ciphertext and the original plaintext will always output the same value if the ciphertext is generated by $F_k(x)$. As D goes with this process round by round, it will get more and more accurate with its guessing.

$\square$

Ex 3.13 **Answer**: Construct F' such that $F'_k(k) = 0^{|k|}$, thus the adversary can get the key k by the reverse permutation, thus the adversary can distinguish the later ciphertext that whether it is generated by $F'_k(k)$ or a random permutation. However, if the adversary only knows one direction of F', it will get no idea of the key, thus it is indistinguishable.

Ex 3.15 Let F be a pseudorandom function, and G a pseudorandom generator with expansion factor $l(n) = n+1$. For each of the following encryption schemes, state whether the scheme has indistinguishable encryptions in the presence of an eavesdropper and wheter it is CPA-secure. In each case, the shared key is a random $k \in 0, 1^n$.

(a) To encrypt $m \in \{0,1\}^{2n+2}$, parse m as $m_1||m_2$ with $|m_1| = |m_1|$ and send $\langle G(k) \ \oplus m_1, G(k+1) \oplus m_2 \rangle$.

Answer: Yes, he has indistinguishable encryptions in the presence of an eavesdropper, but it is not CPA-secure. It is trivial for an adversary to make a CPA attack.

(b) To encrypt $m \in \{0,1\}^{n+1}$, choose a random $r \leftarrow \{0,1\}^n$ and send $\langle r, G(r) \oplus m \rangle$.

Answer: Yes, he has indistinguishable encryptions in the presence of an eavesdropper, and it is CPA-secure.

(c) To encrypt $m \in \{0,1\}^n$, send $m \oplus F_k(0^n)$.

Answer: Yes, he has indistinguishable encryptions in the presence of an eavesdropper. Since in this case, the distribution of $F_k(0^n)$ is indistinguishable from random string of length $n+1$, thus the encryption is just like one-time pad.

However, it is not CPA-secure, which is obvious.

(d) To encrypt $m \in \{0,1\}^{2n}$, parse m as $m_1||m_2$ with $|m_1| = |m_2|$, then choose $r \leftarrow \{0,1\}^n$ at random, and send $\langle r, m_1 \oplus F_k(r), m_2 \oplus F_k(k+1) \rangle$
Answer: Yes, he has indistinguishable encryptions in the presence of an eavesdropper, and it is CPA-secure.

(c) To encrypt $m \in \{0,1\}^n$, send $m \oplus F_k(0^n)$.

Ex 3.21 Let $\Pi_1 = (Gen_1, Enc_1, Dec_1)$ and $\Pi_2 = (Gen_2, Enc_2, Dec_2)$ be two encryption schemes for which it is known that at least one is CPA-secure. The problem is that you don't know is CPA-secure and which one may not be. Show how to construct and encryption scheme Π that is guaranteed to be CPA-secure as long as at least one of $\Pi_1 or \Pi_2$ is CPA-secure. Try to provide a full proof of your answer.
Answer: Our new encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is defined as follows:

Gen: Given input n, it sets $\Pi's$ message space $\mathcal{M}$, key sapce $\mathcal{K}$, and ciphertext space $\mathcal{C}$, $\Pi_1's$ message space $\mathcal{M}_1$, key sapce $\mathcal{K}_1$, and ciphertext space $\mathcal{C}_1$, $\Pi_2's$ message space $\mathcal{M}$, key sapce $\mathcal{K}$, and ciphertext space $\mathcal{C}$ all equal to $\{0,1\}^n$. Then it generates keys for Π_1 and Π_2 as: $k1 \leftarrow \mathsf{Gen}(1^n)$ and $k1 \leftarrow \mathsf{Gen}(1^n)$. Then it sets G to be a pseudorandom generator which takes input $s \in \{0,1\}^l (l < n)$ and outputs string of length n.

Enc: Given keys $k_1, k_2 \in \{0,1\}^n$ and a message $m \in \{0,1\}^n$. First, Enc gets a pseudorandom string $s \in \{0,1\}^n$ using G. It outputs $m_1 = s$ as the plaintext input for Π_1 and $m_2 = s \oplus m$ as the input for Π_2. Then Enc invokes $\mathsf{Enc}_1, \mathsf{Enc}_2$ with keys k_1, k_2 seperately to get ciphertexts $c1, c2$, outputs $c := (c_1, c_2)$.

Dec: Given keys $k_1, k_2 \in \{0,1\}^n$ and a ciphertext $c = (c_1, c_2) \in \{0,1\}^{2n}$. Dec uses $\mathsf{Dec}_1, \mathsf{Dec}_2$ with keys k_1, k_2 to decrypt c_1, c_2 seperately and get two plaintexts m_1, m_2, output $m := m_1 \oplus m_2$.

Next, wo will prove that our scheme is guaranteed to be CPA secure as long as at least one of $\Pi_1 and \Pi_2$ is CPA secure.

Proof. Since at least one of Π_1 or Π_2 is CPA-secure, the adversary can not distinguish both ciphtertexts of Π_1 and Π_2 with non-negligible probability. So the adversary can not recover both m_1 and m_2 with non-negligible probability. Thus the adversary can not recover $m = m_1 \oplus m_2$ with non-negligible probability. $\square$