1. Exercise 5.5: Let $X$ be a Poisson random variable with mean $\mu$, representing the number of errors on a page of this book. Each error is independently a grammatical error with probability $p$ and a spelling error with probability $1 - p$. If $Y$ and $Z$ are random variables representing the number of grammatical and spelling errors (respectively) on a page of this book, prove that $Y$ and $Z$ are Poisson random variables with means $\mu p$ and $\mu (1 - p)$, respectively. Also, prove that $Y$ and $Z$ are independent.

2. Exercise 5.9: Consider the probability that every bin receives exactly one ball when $n$ balls are thrown randomly into $n$ bins.
   a) Give an upper bound on this probability using the Poisson approximation.
   b) Determine the exact probability of this event.
   c) Show that these two probabilities differ by a multiplicative factor that equals the probability that a Poisson random variable with parameter $n$ takes on the value $n$. Explain why this is implied by Theorem 5.6.

3. Exercise 5.12: Suppose that we vary the balls-and-bins process as follows. For convenience let the bins be numbered from 0 to $n - 1$. There are $\log_2 n$ players. Each player randomly chooses a starting location $l$ uniformly from $[0, n - 1]$ and then places one ball in each of the bins numbered $l \mod n$, $l + 1 \mod n$, $l + n/\log_2 n - 1 \mod n$. Argue that the maximum load in this case is only $O(\log \log n / \log \log \log n)$ with probability that approaches 1 as $n \to \infty$.

4. Exercise 5.21: In hashing with open addressing, the hash table is implemented as an array and there are no linked list or chaining. Each entry in the array either contains one hashed item or is empty. The hash function defines, for each key $k$, a probe sequence $h(k, 0), h(k, 1), \ldots$ of table locations. To insert the key $k$, we first examine the sequence of table locations in the order defined by the key’s probe sequence until we find and empty location; then we insert the item at that position. When searching for an item in the hash table, we examine the sequence of table locations in the order defined by the key’s probe sequence until either the item is found or we have found an empty location in the sequence. If an empty location is found, this means the item is not present in the table.
   An open-address hash table with $2^n$ entries is used to store $n$ items. Assume that the table location $h(k, j)$ is uniform over the $2n$ possible table locations and that all $h(k, j)$ are independent.
   a) Show that, under these conditions, the probability of an insertion requiring more than $k$ probes is at most $2^{-k}$.
   b) Show that, for $i = 1, 2, \ldots, n$, the probability that the $i$th insertion requires more than $2 \log n$ probes is at most $1/n^2$.
   Let the random variable $X_i$ denote the number of probes required by the $i$th insertion. You have shown in part (b) that $\Pr(X_i > 2 \log n) \leq 1/n^2$. Let the random variable $X = \max_{1 \leq i \leq n} X_i$ denote the maximum number of probes required by any of the $n$ insertions.
   c) Show that $\Pr(X > 2 \log n) \leq 1/n$.
   d) Show that the expected length of the longest probe sequence is $E[X] = O(\log n)$. 


5. **Exercise 5.22**: Bloom filters can be used to estimate set differences. Suppose you have a set \( X \) and I have a set \( Y \), both with \( n \) elements. For example, the sets might represent 100 our favorite songs. We both create Bloom filters of our sets, using the same number of bits \( m \) and the same \( k \) hash functions. Determine the expected number of bits where our Bloom filters differ as a functions of \( m, n, k \), and \( |X \cap Y| \). Explain how this could be used as a tool to find people with the same taste in music more easily than comparing lists of songs directly.

6. **Exercise 5.26**: Consider Algorithm 5.2, the modified algorithm for finding Hamiltonian cycles. We have shown that the algorithm can be applied to find a Hamiltonian cycle with high probability in a graph chosen randomly from \( G_{n,p} \), when \( p \) is known and sufficiently large, by initially placing edges in the edge lists appropriately. Argue that the algorithm can similarly be applied to find a Hamiltonian cycle with high probability on a graph chosen randomly from \( G_{n,N} \) when \( N = c_1 n \ln n \) for a suitably large constant \( c_1 \). Argue also that the modified algorithm can be applied even when \( p \) is not known in advance as long as \( p \) is at least \( c_2 \ln n / n \) for a suitably large constant \( c_2 \).

7. **Exercise 6.4**: Consider the following two-player game. The game begins with \( k \) tokens placed at the number 0 on the integer number line spanning \([0, n]\). Each round, one player, called the chooser, selects two disjoint and nonempty sets of tokens \( A \) and \( B \). (The sets \( A \) and \( B \) need not cover all the remaining tokens; they only need to be disjoint.) The second player, called the remover, takes all the tokens from one of the sets off the board. The tokens from the other set all move up one space on the number line from their current position. The chooser wins if any token ever reaches \( n \). The remover wins if the chooser finishes with one token that has not reached \( n \).
   a) Give a winning strategy for the chooser when \( k \geq 2^n \).
   b) Use the probabilistic method to show that there must exist a winning strategy for the remover when \( k < 2^n \).
   c) Explain how to use the method of conditional expectations to derandomize the winning strategy for the remover when \( k < 2^n \).