1. What is the sum of the even integers from zero to \( n \) when \( n \) is an even integer?

2. Consider Algorithm 1.12 (Insertion Sort) being applied to \( n \) distinct items where each permutation of the items is equally likely. In Step 5 the algorithm does a comparison of data items. In Steps 2, 5, and 8 the algorithm moves data items.
   
   a. What is the average number of comparisons?

   b. What is the average number of data movements?

3. Consider a process where you roll two six sided dice where the faces are numbered with integers from 1 to 6. Your score is the absolute value of the difference between the two integers that are face up after rolling the dice.
   
   a. What is the average (expected) value of your score?
   
   b. What is the variance of your score?

4. The expression \( x(x - 2) \cdots (x - n + 2) \) is a polynomial in \( x \). Write a formula for the coefficient of \( x^k \) in this polynomial. Use Stirling numbers to the extend that it is appropriate to do so. In other words, if the expression is written in the form \( \sum_i a_{ni} x^i \), what is the value of \( a_{ni} \)?

5. Sometimes it is important to do a rough analysis of an algorithm even before writing it out explicitly. This question asks you to consider a sorting algorithm based on the following ideas. The algorithm will search through the items looking for the smallest item (using ideas similar to 3.2 but avoiding the variable \( y \) so as to reduce data movement (\( y \) can be replaced with \( x_k \)). Once the smallest item is found, it will be interchanged with the first item. Normally, an interchange counts as three data moves (move one item to a temporary location, move the other item to where the one item had been, move the other item where the one item had started out). The outer loop of will repeat the process leaving out the initial small items that have already been moved to their final places.
   
   a. What is the average number of comparisons for this algorithm?

   b. What is the average number of data movements for this algorithm?

   c. Discuss conditions (if any) when this algorithm is preferred over Insertion Sort and when it is not. Consider cases where the items are long (and therefore slow to move) and cases where they are short.