Homework 9: Due Friday, March 13

Note: For each of the counting problems, you must explain your solution. For example, if your answer is a product, describe the sequence of choices you are making and explain where each term comes from. Numerical answers without written justification will receive no credit.

Problem 1: How many 6-letter "words" contain one of the letters A, B, C, D three times and each of the others once?

Problem 2: Suppose that a lottery draws 6 numbers from $\{1, 2, \ldots, 60\}$ without replacement and where order drawn doesn't matter. What percentage of possible lottery numbers have 3 evens and 3 odds?

Problem 3: A local pizza place has three different types of crust, five different meats, and seven different (non meat) toppings. For a given pizza, you can pick any crust, at most 2 meats (so 0, 1, or 2 is possible) and at most 3 toppings (so 0, 1, 2, or 3 is possible). Assume that you can not have double of any topping. How many pizzas are possible?

Problem 4: In class, we talked about the number of paths starting at $(0, 0)$ and ending at (m, n) where each step was either one step north or one step east. How many such paths are there from $(0, 0)$ to $(12, 9)$ which do not go through the point $(5, 4)$? Think of needing to avoid that intersection because of construction.

Problem 5: How many 5-card poker hands have at least one card of every suit?

Problem 6: Let $n, k \in \mathbb{N}$. A sequence of nonnegative integers $(a_1, a_2, \ldots, a_k)$ such that $a_1 + a_2 + \cdots + a_k = n$ is called a *weak composition* of n into k parts. For example $(1, 3, 5, 3)$ is a weak composition of 12 into 4 parts and $(2, 0, 5, 1, 0, 0)$ is a weak composition of 8 into 6 parts. Write a recursive Scheme program that takes two natural numbers n and k as input, and produces the set of all weak compositions of n into k parts. For example, on inputs $n = 3$ and $k = 2$, it should produce

$$'((0\ 3)\ (1\ 2)\ (2\ 1)\ (3\ 0))$$

(although possibly in a different order). If $n = 0$ and $k = 0$, your program should produce $'(())$ (because the empty sequence is technically a weak composition of 0 into 0 parts). If $n > 0$ and $k = 0$, your program should produce $'()$.
Hint: Here's a helpful way to think about this recursively: If $k > 0$, then a weak composition of n into k parts is of one of two types: either it starts with a 0 or it does not. In the former case, if we omit the 0, then we obtain a weak composition of n into $k - 1$ parts. In the latter case, if we decrease the first element by 1, then we obtain a weak composition of $n - 1$ into k parts.