COP 4516 Spring 2025 Week 9 Team Contest #1 Solution Sketches

Digit Solitaire
This is probably the easiest problem in the set. Write a function that takes in an integer n and returns its digit product. This function should peel off each digit of the number one by one via %10 and /10 and multiply them into an accumulator. Then, just call this function over and over again using the return value as the new input until the answer is less than 10.

Extra Set
A triple for loop through all of the cards with a maximum of 1500 cards runs the inner-most code over 1.1 billion times. Even with a 10 second time limit, given the computation that has to occur in the inner-most code, a TLE will result. The key observation is that if you know two of the three cards in a set, you can uniquely determine what the third card has to be. For example, if two cards are (2, 1, 2, 0, 0) and (2, 0, 0, 1, 0), then we see that the first and last attributes for the third card must be the same while the middle three must be different. Using the two cards given and the deduction given, this means the third matching card must be (2, 2, 1, 2, 0). So, the solution is as follows, Store all of the cards in a HashSet. Run a double for loop through all pairs of cards. For each pair, calculate its matching third. Then, check if that matching third is in the set. If so, add 1 to a counter. After adding up everything, divide the final answer by 3 since each triple was discovered 3 times. A triplet of cards at indexes i, j and k with i < j < k is discovered when we look at the ordered pairs (i, j), (i, k) and (j, k).

Say Cheese!
The diagonal is at a 45 degree angle. Using the Pythagorean theorem, that means that we can calculate the width of the frame (not W for the whole picture…) to be the diagonal length divided by the square root of 2. Once we know this value, we can then calculate the necessary lengths of all four trapezoids that form the frame. Basically, the width of the picture along the outside of the frame is just W + Dsqrt(2) and the length along the outside of the frame is L + Dsqrt(2). The area of a trapezoid is just $\frac{1}{2}(b_1 + b_2)h$, where the b's are the bases and h is the height. (We can also decompose the frame into four rectangles and four squares in the corner.)

Hexagram
There are 12! ways to put the numbers in the circles. Unfortunately, 12! is quite big (several hundred million) so straight brute force doesn’t work. The key observation to solve the problem is to recognize that each circle gets added into exactly 2 row sums, so the sum of all six rows equals twice the sum of all the input numbers. This means that for the 6 rows to all have the same sum, let this sum of each row be R and the sum of all the 12 input numbers be S, we must have 6R = 2S. It follows that R = S/3. Thus, before we start our brute force search for solutions, we can precompute what the row sum will be. Then, when we write our brute forcer, whenever we finish any of the six rows, we can immediately check if that row adds to the proper target. If it does not, skip placing that last item in the row and move onto the next one (essentially skip trying a number in a circle that is doomed to fail.)

Knitting

These values are small enough that one can just simulate stitching each row. To figure out what to add to the previous row value, we can just take the row we are on and mod it by the stitching pattern length and use that index into the array storing the cyclic pattern of changes.

Degrees of Separation

This question is asking us to look at the minimum distances between all pairs of nodes in a graph and determine the largest of these values. In graph theory, this is known as the girth of a graph. Since this graph is small, we can run Floyd Warshall's Algorithm on the graph to obtain a 2D array that stores all minimum distances between pairs of vertices. If any of these is "unreachable" then we output DISCONNECTED. Otherwise, we scan the 2D array looking for the largest value in it and output this value, which represents the girth of the graph. Since the graph size is 50 people at most, we can just set each edge between people who don't know each other to a value greater than 50 since all valid shortest distances will be 49 or less.