COP 4516 Kattis Contest #2 Solution Sketches

Problem A: Apaxiaaaaaaaaaaaans!

URL: https://open.kattis.com/problems/apaxiaaans

Loop through the string in bunches. Start at a particular index and loop a different index through the string until you encounter a different letter. Then output the original letter at the starting index once. Update the starting index to where the first different letter is and repeat until you get through the whole string.

Problem B: Server Space

URL: https://open.kattis.com/problems/netthjonaplass

First sort the server locations in order. We know the total width of the first server: it's twice the x-coordinate of the first server. Now, we have the correct left endpoint for the next server and can figure out its width by subtracting the center of it from this new left endpoint and multiplying by two. Continue in this fashion to generate all of the output.

Problem C: Grazed Grains

URL: https://open.kattis.com/problems/grazedgrains

The key to this problem is noticing that you don't have to get very close to the real answer. 10% is a huge amount of error. Also, there are only 10 circles. We can use a probabilistic approach: Create a square in between x = -10, x = 20 and y=-10 and y=20. (Something slightly larger works as well.) Next randomly generate many points (100,000 is enough) within this square. For each randomly generated point, see if it's in any of the circles in the input. Count how many of the randomly generated points are in at least one circle. This will give you an empirical probability, say x/100,000 if you did 100,000 trials of a randomly chosen point being in one of the circles. Finally, take this fraction and multiply it by the area of your square. This gives an approximation for the total area of the squares, which will almost definitely be within 10% of the actual figure. It's important to generate points with coordinates that are equally randomly distributed doubles.

Problem D: Jamboree
URL: https://open.kattis.com/problems/jamboree

If there are n items and m girls, then n – m girls will have to carry 2 items. If we want to minimize the total weight any girl carries, we must give these n – m girls the 2(n – m) lightest items. (If we don't, then we're leaving someone carrying two items to carry a heavier item than necessary.) Then, with the remaining 2m – n heaviest items, we give 1 to each remaining girl. When we assign girls to carry two items, if we aim to minimize the maximum weight any girl is carrying, we must pair the heaviest item with the lightest item. Any other pairing might force that girl to carry more than necessary. (Someone has to carry the heaviest item left and it must be paired with a second item, so greedily pair it with the lightest because we could not ever do better.) Repeating this logic means the second heaviest item is paired with the second lightest item and so on.

Here is a quick example:
n = 10 items
m = 7 girls
sorted list of weights = 2, 3, 4, 4, 6, 9, 11, 12, 12, 14

In this example, we have 10 – 7 = 3 girls carrying the 6 lightest items, so one girl carries the items weight 2 and 9, another 3 and 6 and the last girl 4 and 4. The final four girls carry a single item. So for this example, the answer is 14.

Problem E: Baseball Packs
URL: https://open.kattis.com/problems/baseball

We have 10 players to assign to 10 positions. There are 10! ways to do so at most. Since this is about 3 million and for each assignment it takes us a loop running 10 times to assess if the solution is viable, the algorithm of simply trying each possible assignment will run in time. Thus, run the permutation algorithm assigning players to positions (you can keep one set fixed and permute the other set). To save a bit of time (backtracking), when you try setting a player to a position, only do so if it's one of his potential positions. If any assignment finishes, output "True", otherwise "False".