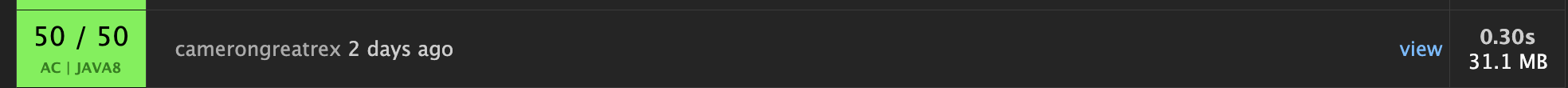
**DMOJ Username:** camerongreatrex

**Fastest Attempt: **

**Test Cases:** I used the two test cases listed on the website along with testing cases such as (1,2,3,50) where the fish are worth low points and the max points are very high. I then checked if those worked by printing loads of combinations, and usually they did. If those both worked I would just submit the project and make sure every test case works. I find that is the most effective way to test if the code works

**Approach to solving the problem**:

I started off using one loop and if statements to calculate the number of possible ways for each fish individually, and the total number of possibilities. I needed to find a way to calculate them at the same time. After that, I basically translated the one for loop and if statements into 3 separate loops and then tried to use them to find the combinations of each fish separately again, but in a systematic way. From there I needed to figure out how to do them all at once. I realized that to do that I’d need to put all the loops together in some way to allow for increasing and decreasing numbers of each fish in a combination. I.e one for loop can only add a value to one fish, and two can only add a value to two fish, etc. I literally tried putting all the for loops together with the same bodies as the separate ones from before and it worked. I then knew that I needed to make sure that the total points where under the max amount of points from before, which means that I needed to add up the value\*count of each fish together for the total points. From there, I knew that if total points are less than the max points I know the combination works and I should print the combo using the current loop variables. I also had to make sure that a combo with 0 total fish was not printed, which is why I added one more check to make sure at least one fish is caught. Finally, I added one to the total amount of possibilities for each time the program runs and finds/prints a combination. Overall, I tried to use a process of elimination method to remove useless logic that I would think of that would not work. Once I had code that somewhat worked, I would improve it and find ways to edit it while keeping the basics of it the same. The biggest jumps were always the hardest to deal with (3 separate loops to one big loop gets more than 3x more complex haha).

**Sources:** No other sources used

**Troubles while making a correct solution:**

I didn’t have too much trouble with this problem, however, I found that adapting 3 single for loops to a triple nested for loop made it hard to trace the program and therefore solve it. I mostly understood how the triple-nested loop worked because of the original 3 separate for loops. I had a small suspicion that trying to literally add the loops together would solve the problem, and for the most part, it did. Once those were added I just needed an if statement with checks to make sure the combination is valid and doesn’t print the wrong numbers.

**Thing that is special:** We out here grinding the DMOJ times

