Dominic Celiano

* Problem 1
  + Add to HW 1
  + Explanation: Set the max number as 1000. Then, iterate through every number from 1 to 1000 and check if the number is divisible by 3 or 5. If it is, add the number to the sum and then print the sum once we are done with the loop.
* Problem 2
  + Add to HW 1
  + Explanation: Define the first two Fibonacci numbers. Then, create a “for” loop that will iterate until the higher number reaches 4,000,000. In the “for” loop, define a new “num\_2” (high number) and check if that number is even (divisible by 2). If it is, add it to the sum, and then print the sum at the end.
* Problem 5
  + Add to HW 3
  + Explanation: We start a “test\_num” variable at a low number (25), which will be incremented by 1 each time and then tested to see if it is divisible by everything between 1 and 20, inclusive. This is tested using a for loop inside the loop in which we are incrementing the test\_num variable. As soon as we find a number that works, we exit all the loops and print the number.
* Problem 6
  + Add to HW 3
  + Explanation: This solution does two things simultaneously. While it loops through the numbers from 1 to 100, inclusive, it both adds the numbers by themselves and adds the squares. At the end of the loop, it squares the non-squared sums and then, from it, subtracts the sum of the squares. It then prints the difference.
* Problem 16
  + Add to HW 4
  + Explanation: This problem sets a variable to be equal to 21000, typecasts that variable into a string, and then iterates through everything character in that string and converts it to an integer as it adds it to a sum. Then, it prints the sum.
* Problem 20
  + Add to HW 4
  + Explanation: First, find the result of 100!. Do this by adding the results of two numbers multiplied together until one of the numbers equals 0. Then, add the results of all the characters together by doing some typecasting in python.
* Problem 25
  + Add to HW 4
  + Explanation: This problem is pretty much the same thing as Problem 2, but instead of testing if a number was even (divisible by 2), I was testing if it had 1,000 digits. I did this by dividing it by 101000-1 until the number was >= 1, meaning that it had 1,000 digits. At that point, I exited the loop and printed the number.