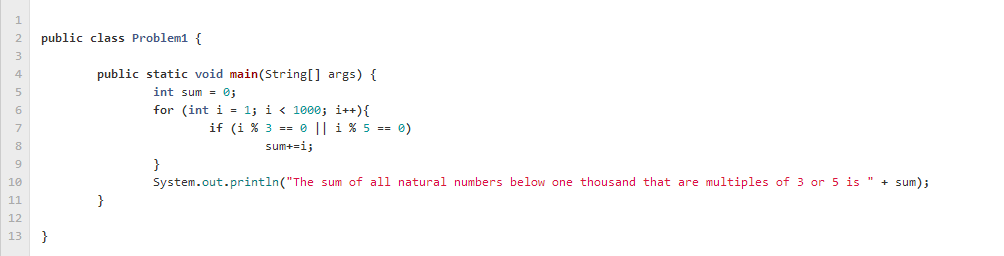
*All of my Project-Euler code is available at* [*http://github.com/fhinson/Project-Euler*](http://github.com/fhinson/Project-Euler) **Francis Hinson**

**Problem 1**

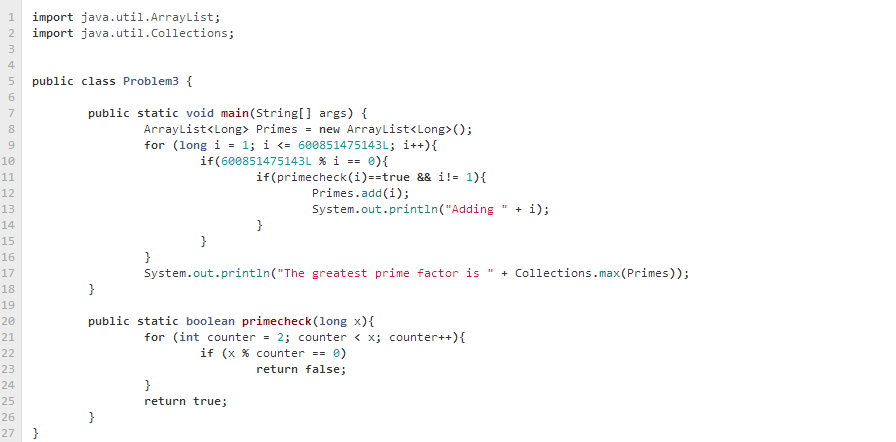
For this problem, using two simple for loops, I added all the numbers under 1000 that are multiples of 3 or 5 to a variable, and then printed that variable.

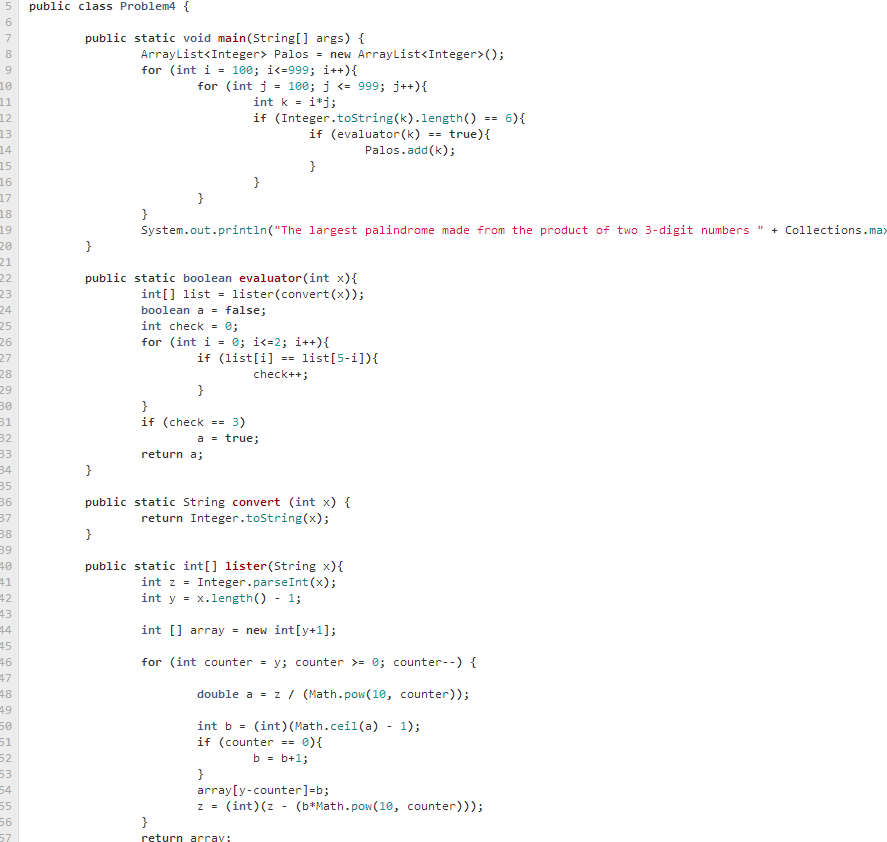
**Problem 2**

For this problem, I created an ArrayList of Fibonacci numbers and using a for-loop, I incremented the sum of the even ones using a simple conditional statement.

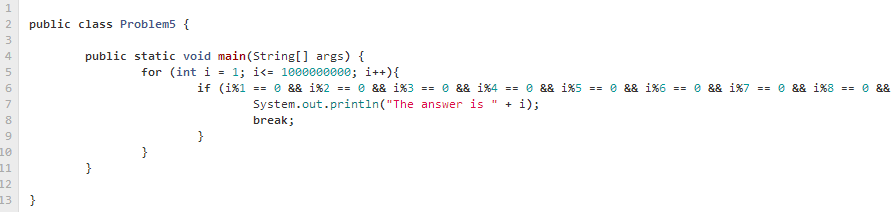
**Problem 3**

For this problem, I defined the number 600851475143 as a long, and then I incremented from 1 to that number and checked for prime factors, adding them to an arraylist. I then printed the greatest element of the arraylist.

**Problem 4**

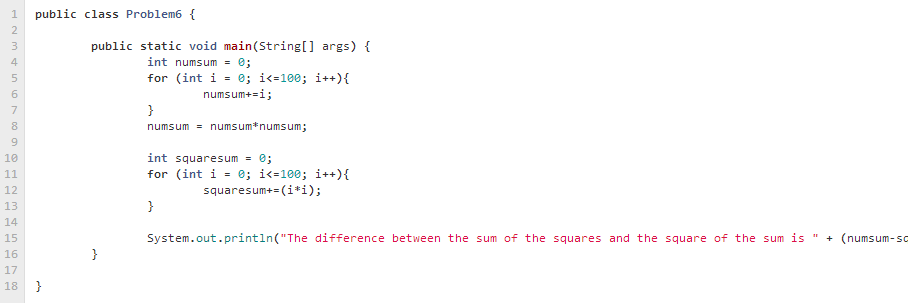
For this problem, I used a method to break a number down into an array of digits. I then had a method that evaluated to see whether that number was indeed a palindrome. I added all the palindromes to an ArrayList, and I printed the maximum value.

**Problem 5**

For this problem, I simply searched for the smallest number evenly divisible by 1-20 with a conditional statement.

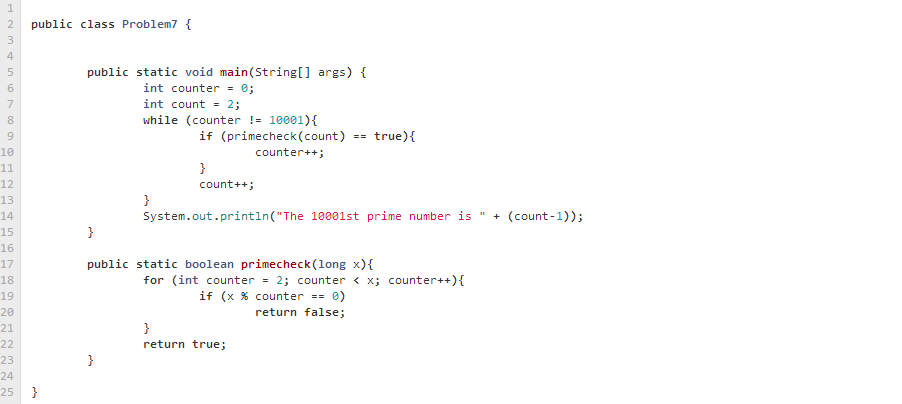
**Problem 6**

For this problem I simply dealt with two for-loops, one which calculated the sum of the squares and another which calculated the square of the sums. I then took the difference between them to arrive at the answer.



**Problem 7**

For this problem, I simply used a method that checks whether a number is prime, and looped through a lot of numbers until my counter arrived at 10001.

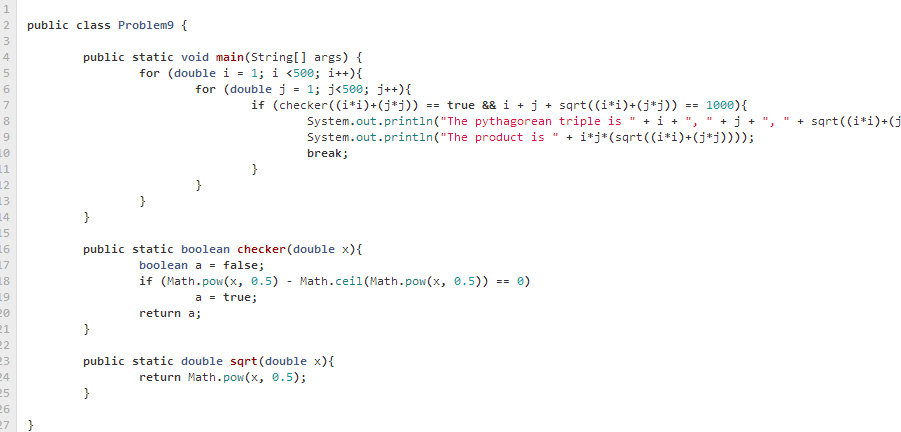


**Problem** 8

For this problem, I put the 1000 digit number into an array of digits, and I added the products of 5 consecutive numbers through a for loop to an arraylist. I then returned the maximum value of that arraylist.

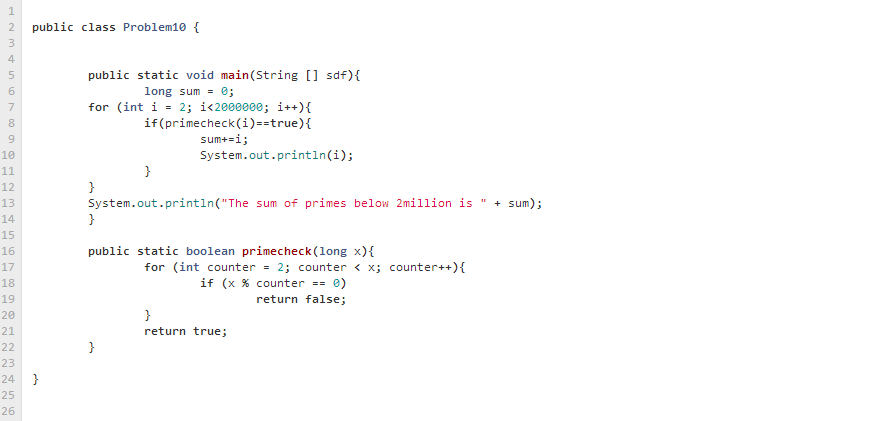
**Problem 9**

For this problem, I used a double for loop and incremented through values until I found the Pythagorean triple that equated to 1000.



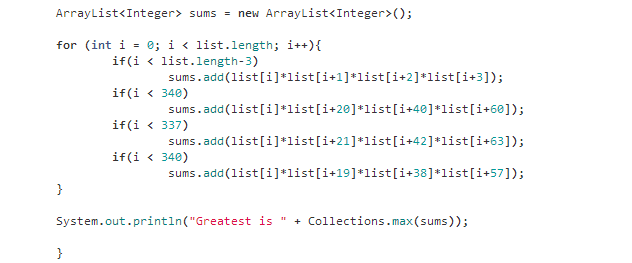
**Problem 10**

For this problem, I had a simple method to check if the number was prime. Then, I incremented through numbers under 2million, adding the primes along the way. The result was simply the total.



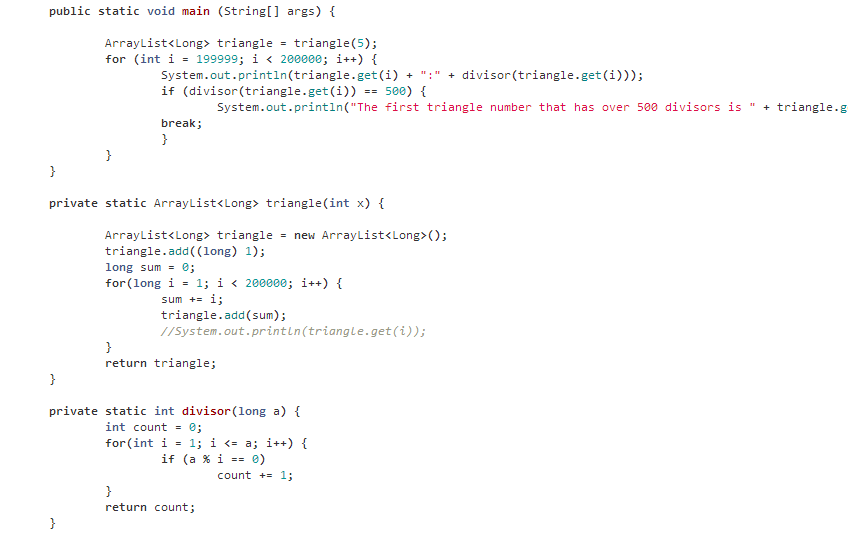
**Problem 11**

For this problem, I put all of the numbers into an array and started analyzing groups of 4 using for loops. I had 4 distinct conditionals to handle left, right, diagonal right, and diagonal left.



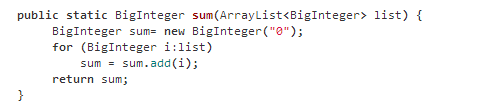
**Problem 12**

For this problem, I had a method which created my triangular numbers. Then I had another method which calculated the number of divisors a number has. I incremented through the triangular numbers using a for loop until I found the one with the number of divisors over 500.



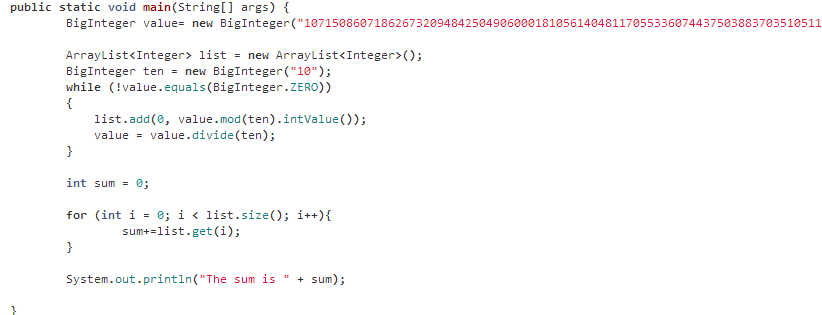
**Problem 13**

For this problem, I simply summed the large numbers using the Java BigInteger class. I then found the first 10 digits of the sum easily.



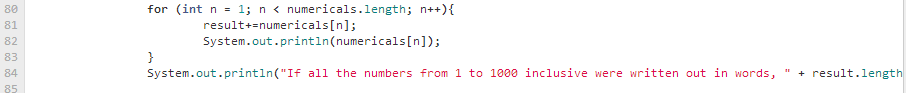
**Problem 16**

For this problem, I contained 2^1000 in a BigInteger, and then I broke the number down into an array of digits. Using a for-loop, I easily summed the digits.



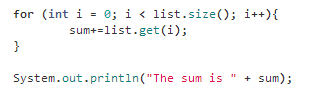
**Problem 17**

For this problem, I essentially created a small dictionary for the program to refer to of numbers in numerical format. I then designed an algorithm to combine certain numericals. For example, it would take 42 and make that 40 + 2. Using a lot of looping structures, I arrived at the total length.



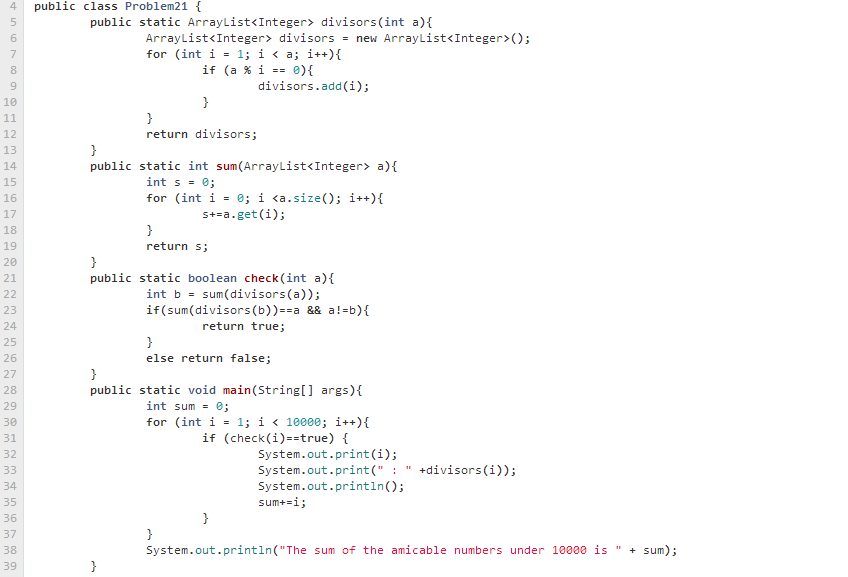
**Problem 20**

For this problem, I stored 100! in a BigInteger and then I broke the number down into an array of digits and simply added the digits using a for loop.



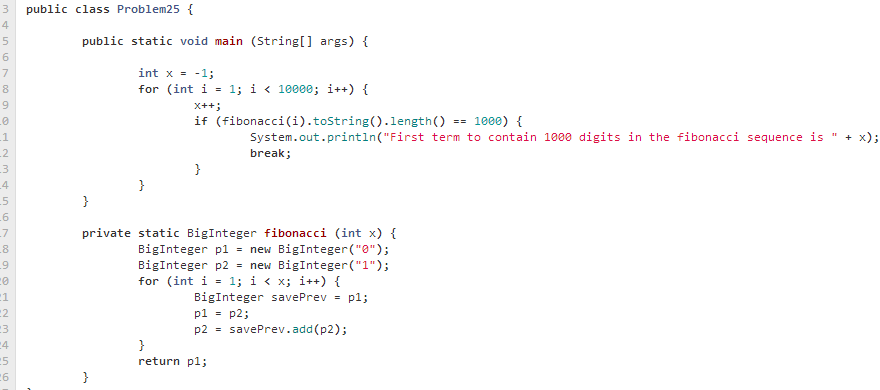
**Problem 21**

For this problem, I created a method to sum the divisors of a number. I then looped through several numbers to find amicable pairs, and I summed them.



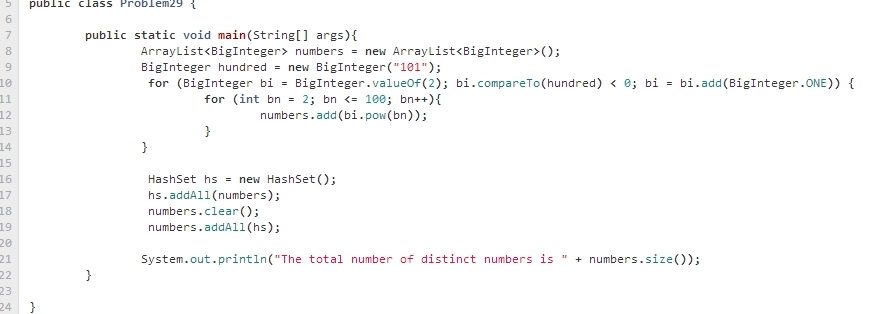
**Problem 25**

For this problem, I had a method that uses BigIntegers to generate Fibonacci numbers. I then used a forloop to increment through these numbers, parsed them to strings to find their lengths, and found the one with the length of 1000.



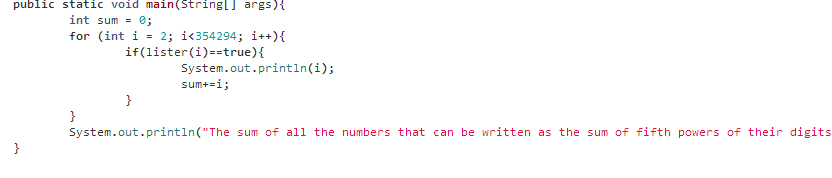
**Problem 29**

For this problem, I simply used a double for loop and stored all of the values of a^b in an ArrayList. I then dumped all of the values of the arraylist into a hashset, which removed duplicates. I then obtained the total number of values.

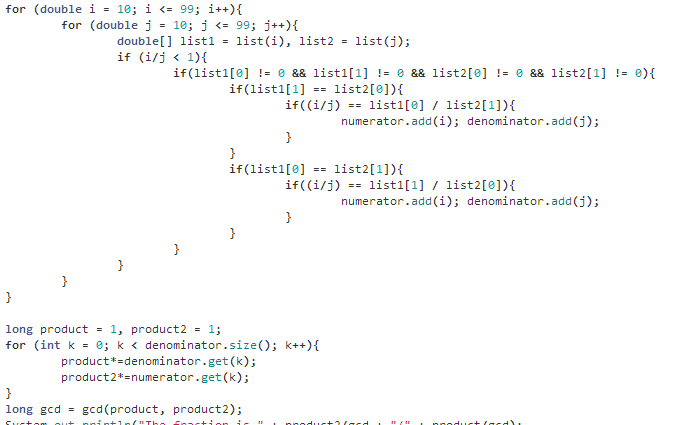


**Problem 30**

For this problem, I created a method which breaks a number into digits and sums the fifth power of the digits and determines whether they equal the original number. I then used a for loop to find such numbers, and I took the sum of them.

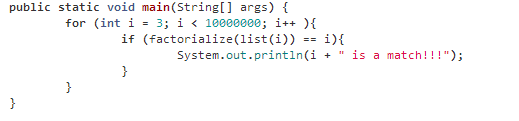


**Problem 33**

For this problem, I used a method to break numbers into digits, and then I started removing digits from the fractions and making evaluations using lots of looping and conditionals.

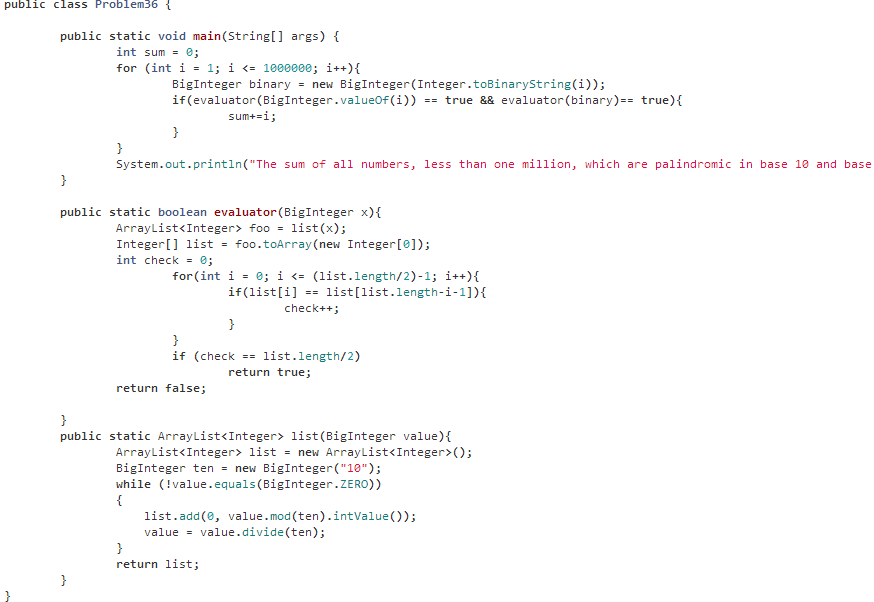
**Problem 34**

For this problem, I broke a number down into an array of digits, and summed the factorials of the number. I looped through several numbers to find my matches, and summed them. I used a recursive method to factorial the numbers.



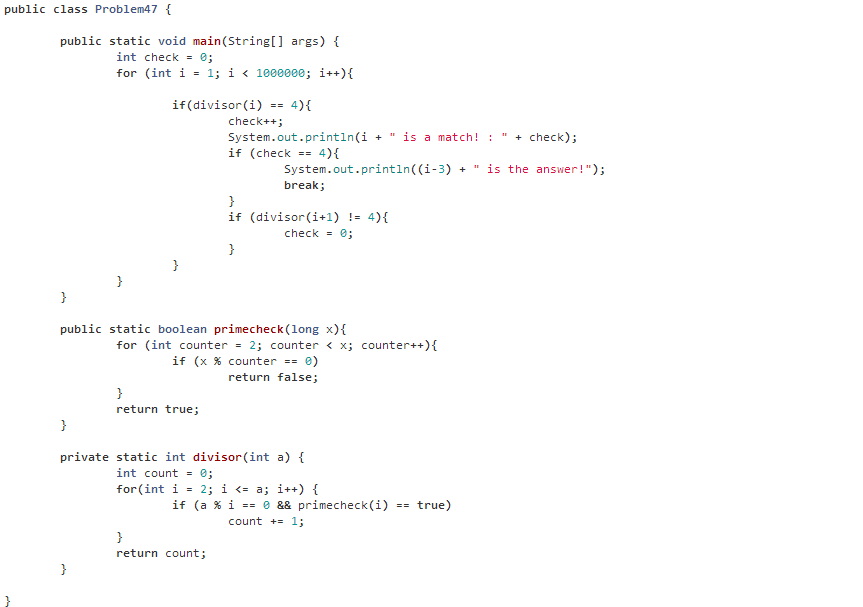
**Problem 36**

For this problem, I converted the number to an array of digits, broke the array in half, checked if the 1st value corresponds with the last and so on until it reaches the middle. If there were matches, it increments a counter, and if the number is indeed a palindrome, the number of matches should be equal to half the length of the number. I applied this in a looping structure to regular numbers and their binary equivalent to find the sum.



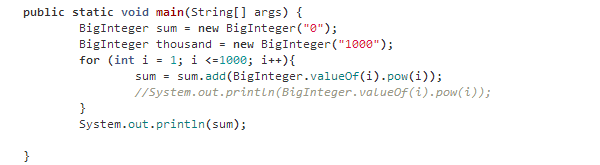
**Problem 47**

For this problem, I wrote methods to find the number of divisors and whether they are prime. I then used a for loop to find 4 consecutive numbers that have those traits.



**Problem 48**

For this problem, I simply used the BigInteger class to make this large summation. I then easily obtained the last 10 digits.



**Problem 50**

For this problem, I used a lot of looping structures to search for the longest chain of primes that sum to another prime. I used a counter point system to determine the maximum length.

