**Gödel Numbering Assignment Write-Up**

I was rather excited for this assignment as I didn’t anticipate it taking very long due to previous code I had for finding unique prime factors and determining if a number was prime. While I did encounter some unexpected issues that meant I was unable to complete it in class, these issues were resolved relatively soon after with troubleshooting and input from Dr. Reinhart. I’ve been trying to code by best practice more often and decided to split things into classes rather than a spaghetti file. This resulted in a reused PrimeNumber class from COMP 220 and a GodelNumbering class with a separate class for running test cases. The first issue didn’t show up until I was using the assigned inputs, at which point the Gödel Number my code was printing out did not match the math it was doing. After adding print statements to each multiplication during the calculation of the Gödel Number, it appeared that my result was always a power of 2 greater than what it should be. When consulting with Dr. Reinhart he realized that this was an issue of mismatched types between my numbers and Math.pow(). Once I cast the result of Math.pow() to a long, everything began working smoothly. The other issue was an absurdly long runtime when generating the reverse mapped sequence. I added print statements at the completion of every method and found that my method for finding unique prime factors was slowing everything down. After tinkering for a bit, I realized that I was only dividing by each factor one, which would generate the factors but only at an incredibly slow pace. If I instead divided by each found factor until it was no longer a factor, I reduced the time run time exponentially. It now completes the test sequence in apparently between 2-3 milliseconds down from minutes before.

Text

Description automatically generated

Third line is the runtime print.