Neil Formosa

Written task, task 2

Is your PRNG implementation correct? Base your answer on the results obtained for code requested in point A above.

Yes the PRNG implementation is correct. The test in the Program.cs file created 100 random numbers between 1 and 1000 using the SplitMix64 algorithm. All of the numbers were in the correct range and they weren’t in any obvious order (like sorted from smallest to largest or largest to smallest). This shows that the PRNG is working properly and producing numbers that look random.

Is your PRNG implementation intractable? Base your answer on the Log-Log Graph produced in point B above.

No my PRNG implementation is not intractable. Based on the Log-Log Graph the time taken to generate random numbers grows in a straight line with the input size meaning it handles large numbers well and doesn’t get stuck or take too long.