1) Fast primality testing

Many internet security protocols require fast generation of large numbers that are prime, or that are "probably prime".

You will implement in code both the Agrawal-Kayal-Saxena (AKS) polynomial-time primality testing algorithm from 2002, which is deterministic, and the Miller-Rabin primality test from 1980, which is probabilistic. You will compare their theoretical time complexities, as well as their speed in practice.

In 2002, Agrawal, Kayal and Saxena published the first ever deterministic algorithm of polynomial time complexity for determining whether an arbitrary given number is prime, whose proof of correctness does not depend on the generalised Riemann hypothesis, nor on any other mathematical conjecture. This answered, in the affirmative, the long-standing question of whether the primality decision problem was in the class "P":

https://people.csail.mit.edu/vinodv/COURSES/MAT302-S13/AKSpaper.pdf

In this project you will:

a) Implement the AKS algorithm in code.
b) Implement the Miller-Rabin probabilistic primality test in code.
c) Carry out empirical measurements of the time complexity of both programs and plot the results.

2) Compressed Sensing

This theory, from 2002, uses randomness to "beat" the famous Nyquist-Shannon lower bound on the number of samples need to reconstruct a signal, at least in certain (commonly occurring) cases. It caused a breakthrough in medical imaging: It meant that Computed Tomography (CT) scanners could be made safe enough for use on young children. Previously, to obtain a high-quality image required exposing the patient to a dose of radiation that was too dangerous for children. You will write code to demonstrate the basic idea. See here:

https://youtu.be/A8W1I3mtjp8

3) Programming the Graphics Processing Unit (GPU) using OpenGL

OpenGL is a widely used Application Programming Interface (API) for programming graphics. It is commonly used to control the GPU in order to achieve hardware-accelerated rendering. You will control the GPU using OpenGL in C or C++ to generate fast visualisations of interesting graphics. Possible subjects include the Platonic solids (the tetrahedron, cube, octahedron, dodecahedron and icosahedron), 3d fractals, Coulson's coloured tiling of 3d space from 2002 which proves that the chromatic number of space is at most 15, or anything else that you would like to do.