1 Mind the gap (20 points)

The difference between two consecutive prime numbers is called a prime gap. Write a function which returns all pairs of primes less than a given number `n` and separated by a gap `d`.

You should submit a file named *prime_gap.m* to the CMS.

2 Frequency of letters (20 points)

Write a function which returns the number of occurences of letters `[a-z]` in a given string.

You should submit a file called *letter_freq.m* to the CMS.

3 Polynomials (40 points)

We can represent a polynomial using an array by storing its coefficients. For example the polynomial $p(x) = a_n x^n + .. + a_2 x^2 + a_1 x + a_0$ can be represented with the array `[a0 a1 a2 ... an]`.

(a) Write a function which evaluates a polynomial at a given value of x. The inputs of the function should be a coefficient array and `x`. (10 points).

(b) Modify your evaluation function to return values at multiple points; i.e. assume x values are also given as an array, and the output is an array of values. (10 points).

(c) Write a function which multiplies two polynomials. Your function should return an array of coefficients corresponding to the product polynomial. (20 points).

(d) Write a test script which tests your polynomial functions for various test cases. (10 points).

You should submit four files named *poly_eval.m*, *poly_eval_array.m*, *poly_multiply.m* and *poly_test.m* to the CMS.

4 Astroid (20 points)

In this problem we will explore a numerical experiment to compute the area of an astroid curve. Let's say we have a square shaped board with side length 2, centered at the origin. On this board an astroid curve is drawn with $a = 1$. When we randomly pick N points on the board such that the point density (points per area) is almost constant everywhere on the board, we can compute the ratio of the area of the astroid to that of the enclosing square by counting the number of points falling inside them.

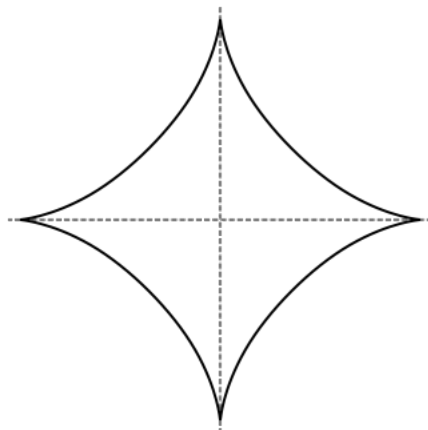

Figure 1: The curve defined by equation $x^{\frac{2}{3}} + y^{\frac{2}{3}} = a^{\frac{2}{3}}$

(a) First you will generate $2N$ random numbers, between -1 and 1 using `rand` function. N of them will be stored in an array to represent x-coordinates, and the other N will be stored in an array to represent y-coordinates. (10 points).

(b) Then you will count the number of points inside the astroid. What is your estimate for the area of the astroid? (10 points).

You should upload a script file named *astroid.m* to the CMS.

5 Soccer Ball Toss (♡ 40 points)

Your boss at *WhatAreTheOdds* asked you to compute the percentage of error made by assuming the soccer ball as a truncated icosahedron. But, you don't know how to compute the exact value using spherical trigonometry. You have told about this problem to your friend who works for a consulting company called *LightAndMagic*. He told you that he'd worked on a project for a Halloween Party once, for which they had to compute the ratio of light coming out of a carved pumpkin. He said they could provide you with the codes they had written for that project. He gave you helper functions which returns the vertices of a truncated icosahedron and a triangulation.

You can think of the lines on the soccer ball as the shadows of the edges of a truncated icosahedron that fits inside the sphere and when there is an isotropic light source at the center. Similar to the Astroid problem, the ratio of the pentagons and hexagons can be found by shooting random rays that hits the surface of the sphere and forms a constant density of points.

(a) Write a function which returns 1 if a given ray hits a triangle, and 0 if it doesn't. You can assume your ray starts at the origin and the direction is specified using 3 components of a vector. The vertices of the triangle are also given as inputs 3 elements arrays. (20 points)

(b) Using the triangulation of hexagons and pentagons, count the number of rays that hits black and white regions on the sphere. What are the probabilities P_B, P_W? (20 points)

You should upload two files named *ray_intersects_triangle.m* and *estimate_probabilities.m* to the CMS. This is a bonus question.