PROGRAMMING

150 Challenge Problems

Loops • Conditionals • Mathematics • Algorithms

Instructor: Thomas Basyal

AI/ML Engineering Department

CONSTRAINTS:

- Use ONLY: `if-else`, `for`, `while`, basic variables
- NO functions, lists, tuples, sets, dictionaries
- NO built-in mathematical functions
- Pure algorithmic implementations required

February 6, 2026

Contents

1 Fundamental Operations & Calculators

Question 1

Create a scientific calculator using nested if-else statements that performs: addition, subtraction, multiplication, division, modulo, integer power (x^n), nth root, and percentage calculations. Handle all edge cases including division by zero and invalid operations.

Question 2

Implement factorial calculation for numbers up to 20 using only while loops. Display each multiplication step. Verify your result by calculating the sum of divisors of the factorial.

Question 3 HARD

Calculate the GCD of three numbers using only nested while loops and the Euclidean algorithm. Extend this to find the LCM of the same three numbers without using any multiplication (use only addition).

Question 4

Write a program to determine if a number is prime. If prime, find the next prime number. If composite, find all its prime factors using trial division with optimization (only check up to sqrt(n)).

Question 5 HARD

Implement the extended Euclidean algorithm using while loops to find coefficients x and y such that ax + by = gcd(a,b). Display all intermediate steps.

Question 6

Calculate $a^b \mod m$ efficiently using binary exponentiation method (modular exponentiation) with only loops and conditionals. Handle cases where b is very large (up to 10000).

Question 7 EXTREME

Convert a decimal number to any base (2 to 36) using only loops. Then convert it back to decimal and verify. Use if-else to map digits 10-35 to letters A-Z.

Question 8

Implement a program that checks if a given number is a perfect square without using square root function. Use binary search approach with while loop.

Question 9 HARD

Find all Armstrong numbers (Narcissistic numbers) between 1 and 1000000. For each number, check if the sum of each digit raised to the power of the number of digits equals the number itself.

Question 10

Determine if a number is: Perfect (sum of divisors = number), Abundant (sum ¿ number), or Deficient (sum ¡ number). Find all perfect numbers less than 10000.

2 Advanced Number Theory & Cryptography

Question 11 HARD

Implement the Sieve of Eratosthenes without using arrays - use nested loops to mark and identify primes up to n=10000. Count twin primes (primes differing by 2).

Question 12 EXTREME

Calculate Euler's totient function (n) by counting numbers coprime to n. Then verify Euler's theorem: $a^{\phi(n)} \equiv 1 \pmod{n}$ for coprime a and n.

Question 13 HARD

Find all Pythagorean triplets (a, b, c) where $a^2 + b^2 = c^2$ and $a + b + c = 1000$. Use primitive triplet generation formulas with loops.

Question 14

Implement the Chinese Remainder Theorem for two congruences using only loops and conditionals. Solve: $x \equiv a_1 \pmod{m_1}$ and $x \equiv a_2 \pmod{m_2}$.

Question 15 EXTREME

Generate the Fibonacci sequence modulo m up to the nth term. Find the Pisano period (the period of Fibonacci sequence modulo m) for given m.

Question 16 HARD

Implement Fermat's primality test: check if $a^{p-1} \equiv 1 \pmod{p}$ for multiple random values of a. Identify Carmichael numbers (pseudoprimes).

Question 17

Find the multiplicative order of an element a modulo n (smallest k where $a^k \equiv 1 \pmod{n}$). Determine if a is a primitive root modulo n.

Question 18 EXTREME

Implement the Pollard's rho algorithm for integer factorization using only loops. Find the smallest non-trivial factor of a composite number.

Question 19 HARD

Calculate the Jacobi symbol $\left(\frac{a}{n}\right)$ using only loops and conditionals. Use properties of the Jacobi symbol to optimize computation.

Question 20

Find all solutions to the linear Diophantine equation $ax + by = c$ where $|x|, |y| < 100$ using nested loops. Verify that gcd(a,b) divides c.

3 Complex Patterns & Sequences

Question 21

Print a hollow diamond pattern with numbers where outer edge has sequential numbers and inner hollow shows calculation $n \times row$.

Question 22 HARD

Generate Pascal's triangle up to row n. For each element, calculate $\binom{n}{k}$ without using factorial function directly. Highlight prime numbers.

Question 23

Create a spiral matrix of size n×n filled with numbers 1 to n^2 in spiral order using only loops and directional conditionals.

Question 24 EXTREME

Print a fractal-like pattern (Sierpinski triangle approximation) using nested loops with if-else to determine positions based on binary representation of coordinates.

Question 25 HARD

Generate Floyd's triangle but with prime numbers only. Each row should contain the next consecutive primes.

Question 26

Create a zigzag pattern where numbers increase diagonally. Implement using loops with direction-changing logic based on row and column parity.

Question 27 EXTREME

Print a hexagonal pattern of numbers where each cell value is the sum of its coordinates in hexagonal coordinate system. Size parameter: radius r.

Question 28 HARD

Generate a pattern showing binary representations arranged in a pyramid, where row n shows numbers 1 to n in binary.

Question 29

Create a "magic square" checker: verify if a given 3×3 arrangement (input as 9 separate variables) forms a magic square where all rows, columns, and diagonals sum to the same value.

Question 30 HARD

Print a butterfly pattern using nested loops where the left wing shows ascending sequences and right wing shows descending, meeting at the center.

4 Calculus & Mathematical Analysis

Question 31 HARD

Calculate e^x using Taylor series: $\sum_{n=0}^{\infty} \frac{x^n}{n!}$ up to relative error ¡ 10^{-6}. Use while loop to check convergence.

Question 32 EXTREME

Implement numerical differentiation: approximate $f'(x)$ for $f(x) = x^4 - 3x^3 + 2x$ using central difference formula with decreasing h values until convergence.

Question 33 HARD

Calculate $\sin(x)$ using Taylor series: $\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!}$. Compare with $\cos(x)$ series calculation at same x.

Question 34

Approximate π using Monte Carlo method simulation: generate random points in unit square, count those inside quarter circle, estimate $\pi \approx 4 \times \frac{\text{inside}}{\text{total}}$. Use LCG for random number generation.

Question 35 EXTREME

Implement Simpson's 1/3 rule for numerical integration of $f(x) = \sin(x) \cdot e^{-x}$ from a to b with n subintervals. Compare with trapezoidal rule result.

Question 36 HARD

Calculate $\ln(x)$ for x near 1 using series: $\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + ...$ For other x, use logarithm properties with loops.

Question 37

Find the root of $f(x) = x^3 - 2x - 5 = 0$ using bisection method. Iterate until $|f(x)| < 10^{-5}$. Display all intermediate approximations.

Question 38 EXTREME

Implement Newton-Raphson method to find $\sqrt[n]{a}$ by solving $x^n - a = 0$. Use numerical derivative approximation. Compare convergence rate with binary search.

Question 39 HARD

Calculate the Riemann zeta function $\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$ for integer s ¿ 1 up to desired accuracy. Verify $\zeta(2) \approx \frac{\pi^2}{6}$.

Question 40

Approximate the derivative $\frac{d^2 f}{dx^2}$ (second derivative) using finite differences for $f(x) = e^x \cos(x)$ at a given point.

Question 41 EXTREME

Implement Romberg integration: use Richardson extrapolation on trapezoidal rule results to achieve higher accuracy for $\int_0^1 \frac{4}{1+x^2} dx$ (should give).

Question 42 HARD

Calculate partial sum of Basel problem: $\sum_{n=1}^{N} \frac{1}{n^2}$ for large N. Estimate the error from actual value $\frac{\pi^2}{6}$.

Question 43

Evaluate the infinite product: $(1 + \frac{1}{2})(1 + \frac{1}{4})(1 + \frac{1}{8})...$ up to n terms. Analyze convergence pattern.

Question 44 EXTREME

Compute the Madhava-Leibniz series for π: $\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \ldots$. Apply Shanks transformation to accelerate convergence.

Question 45 HARD

Implement the secant method to solve $\tan(x) = x$ in the interval $[4, 5]$. Compare iterations needed vs bisection method.

5 Matrix Operations & Linear Algebra

Question 46

Multiply two 3×3 matrices using nested loops. Store input matrices as 9 separate variables each (a11, a12, ..., a33, b11, b12, ..., b33).

Question 47 HARD

Calculate the determinant of a 4×4 matrix using cofactor expansion. Implement recursively using loops (manual recursion with state tracking).

Question 48

Find the inverse of a 3×3 matrix using adjugate method: $A^{-1} = \frac{1}{\det(A)} \text{adj}(A)$. Calculate cofactors using loops.

Question 49 EXTREME

Implement Gaussian elimination with partial pivoting on a 4×4 augmented matrix to solve a system of linear equations. Handle row swaps using conditionals.

Question 50 HARD

Calculate matrix power A^n for 2×2 matrix A using binary exponentiation method with matrix multiplication.

Question 51

Compute the trace, Frobenius norm, and condition number estimate for a 3×3 matrix.

Question 52 EXTREME

Perform LU decomposition of a 3×3 matrix (A = LU) using Doolittle's method with only loops and conditionals.

Question 53 HARD

Find eigenvalues of a 2×2 matrix by solving characteristic equation $\det(A - \lambda I) = 0$ using quadratic formula implemented with loops.

Question 54

Implement the power iteration method to find the dominant eigenvalue and eigenvector of a 3×3 matrix.

Question 55 EXTREME

Calculate the QR decomposition of a 3×3 matrix using Gram-Schmidt orthogonalization process with only loops.

Question 56 HARD

Solve a tridiagonal system of equations using Thomas algorithm (specialized Gaussian elimination) for n=5 equations.

Question 57

Compute the matrix exponential e^A for a 2×2 matrix using Taylor series: $e^A = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots$

Question 58 HARD

Implement Jacobi iteration method to solve a 3×3 system of linear equations. Iterate until convergence (residual ¡ 10^{-4}).

Question 59 EXTREME

Calculate the singular value decomposition (SVD) approximation for a 2×2 matrix using power iteration and deflation.

Question 60

Find the rank of a 4×4 matrix by reducing to row echelon form using loops and counting non-zero rows.

6 Vector Calculus & Geometry

Question 61

Calculate the dot product, cross product, and scalar triple product of three 3D vectors given as separate x, y, z components.

Question 62 HARD

Find the angle between two vectors in 3D space, their projection lengths, and determine if they are orthogonal, parallel, or neither.

Question 63

Given three points in 3D, determine if they are collinear, and if not, find the area of the triangle they form using Heron's formula.

Question 64 EXTREME

Calculate the volume of a tetrahedron formed by four 3D points using scalar triple product. Verify using determinant method.

Question 65 HARD

Find the equation of a plane passing through three non-collinear points. Express in form $ax + by + cz = d$.

Question 66

Calculate the distance from a point to a line in 3D space using vector projection methods.

Question 67 EXTREME

Implement Rodrigues' rotation formula to rotate a 3D vector around an arbitrary axis by angle using only loops and trigonometry.

Question 68 HARD

Find the intersection point of two lines in 3D space (if it exists) using parametric equations solved with loops.

Question 69

Calculate the centroid, circumcenter, and orthocenter of a triangle given three 2D vertices.

Question 70 EXTREME

Determine if a point lies inside a convex polygon (up to 10 vertices) using ray casting algorithm with loops.

7 Combinatorics & Discrete Mathematics

Question 71 HARD

Calculate $\binom{n}{r} = \frac{n!}{r!(n-r)!}$ without computing factorials directly. Use the multiplicative formula with loops to avoid overflow.

Question 72

Generate the nth Catalan number using the formula: $C_n = \frac{1}{n+1}\binom{2n}{n}$. Verify with recursive definition.

Question 73 EXTREME

Calculate the number of derangements (permutations where no element is in its original position) of n objects using inclusion-exclusion principle.

Question 74 HARD

Find the number of ways to partition integer n into exactly k parts using loops to generate and count valid partitions.

Question 75

Calculate Bell numbers B_n (number of partitions of a set with n elements) up to $n = 10$ using triangular array method.

Question 76 EXTREME

Implement the Inclusion-Exclusion Principle to count integers from 1 to n divisible by at least one of the primes: 2, 3, 5, 7.

Question 77 HARD

Calculate Stirling numbers of the second kind $S(n, k)$ (number of ways to partition n objects into k non-empty subsets) using recurrence relation.

Question 78

Find the number of surjective functions from a set of size m to a set of size n using Stirling numbers.

Question 79 EXTREME

Calculate the number of integer solutions to $x_1 + x_2 + x_3 + x_4 = n$ where each $x_i \geq 0$ using stars and bars method.

Question 80 HARD

Implement Burnside's lemma to count distinct necklaces with n beads of 2 colors under rotational symmetry.

8 Numerical Methods & Root Finding

Question 81

Solve the system of nonlinear equations using Newton's method in 2D: $x^2 + y^2 = 5$ and $xy = 2$. Calculate Jacobian numerically.

Question 82 EXTREME

Implement Muller's method to find complex roots of polynomial $x^3 - 1 = 0$ using only real arithmetic with loops.

Question 83 HARD

Use fixed-point iteration to solve $x = \cos(x)$. Analyze convergence and compare with different starting points.

Question 84

Implement Steffensen's method (acceleration of fixed-point iteration) to solve $e^x = 3x$ with faster convergence than Newton's method.

Question 85 EXTREME

Solve the initial value problem $\frac{dy}{dx} = x + y$, $y(0) = 1$ using Runge-Kutta 4th order method for $x \in [0, 2]$.

Question 86 HARD

Implement the false position method (Regula Falsi) for finding root of $x \cdot \log_{10}(x) - 1.2 = 0$. Track convergence rate.

Question 87

Use Euler's method to solve $\frac{dy}{dx} = -2xy$, $y(0) = 1$ and compare with exact solution $y = e^{-x^2}$.

Question 88 EXTREME

Implement adaptive quadrature (recursive trapezoidal rule) to integrate $\int_0^\pi \sin(x)dx$ with automatic step size refinement.

Question 89 HARD

Solve boundary value problem: $y'' + y = 0$, $y(0) = 0$, $y(\pi) = 0$ using finite difference method with n=10 grid points.

Question 90

Implement Brent's method combining bisection, secant, and inverse quadratic interpolation for robust root finding.

9 Advanced Algorithms & Optimization

Question 91 EXTREME

Implement the Simplex algorithm (one iteration) to solve linear programming problem: Maximize $z = 3x + 2y$ subject to $x + y \leq 4$, $2x + y \leq 5$, $x, y \geq 0$.

Question 92 HARD

Use gradient descent to minimize $f(x, y) = x^2 + 4y^2$ starting from point (4,4). Calculate gradient numerically using finite differences.

Question 93

Implement Dijkstra's algorithm simulation on a 5×5 grid where each cell has a cost. Find shortest path from top-left to bottom-right using only loops.

Question 94 EXTREME

Solve the 0/1 Knapsack problem for n=6 items using dynamic programming approach with nested loops (simulate DP table with variables).

Question 95 HARD

Implement the Floyd-Warshall algorithm to find all-pairs shortest paths in a weighted graph represented as 5×5 adjacency matrix.

Question 96

Use the Fast Fourier Transform (FFT) butterfly structure to multiply two polynomials of degree 3 (4 coefficients each) using only loops.

Question 97 EXTREME

Implement the Karatsuba algorithm for fast multiplication of two large numbers (represented as separate digits) using divide-and-conquer with loops.

Question 98 HARD

Solve the Tower of Hanoi problem for n=5 disks by simulating the recursive solution using loops and state variables (manual stack).

Question 99

Implement bubble sort, selection sort, and insertion sort on the same dataset (10 numbers stored as separate variables). Count comparisons for each.

Question 100 EXTREME

Use the Miller-Rabin primality test with k=5 iterations to determine if a large number (up to 10^9) is probably prime.

10 Special Mathematical Functions

Question 101 HARD

Calculate the Gamma function $\Gamma(n)$ for non-integer values using Lanczos approximation with loops.

Question 102

Evaluate the Beta function $B(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1}dt$ using numerical integration.

Question 103 EXTREME

Compute the error function $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}dt$ using Simpson's rule with adaptive refinement.

Question 104 HARD

Calculate Bessel function of the first kind $J_0(x)$ using series expansion: $J_0(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{(m!)^2} \left(\frac{x}{2}\right)^{2m}$.

Question 105

Evaluate the incomplete gamma function $\gamma(a, x) = \int_0^x t^{a-1}e^{-t}dt$ for integer a using numerical integration.

11 Differential Equations & Dynamical Systems

Question 106 EXTREME

Simulate the Lorenz system: $\frac{dx}{dt} = \sigma(y - x)$, $\frac{dy}{dt} = x(\rho - z) - y$, $\frac{dz}{dt} = xy - \beta z$ for 1000 time steps using Euler method.

Question 107 HARD

Solve the predator-prey (Lotka-Volterra) equations using RK4 method and track population dynamics over time.

Question 108

Implement the modified Euler method (Heun's method) to solve $y' = y - x^2 + 1$, $y(0) = 0.5$ on interval [0,2].

Question 109 EXTREME

Solve the second-order ODE $y'' + 2y' + 2y = \sin(x)$ by converting to system of first-order ODEs and using RK4.

Question 110 HARD

Implement the shooting method to solve two-point boundary value problem: $y'' = -y' - y + \ln(x)$, $y(1) = 0$, $y(2) = \ln(2)$.

12 Complex Number Operations

Question 111

Perform complex arithmetic: addition, subtraction, multiplication, division of complex numbers $a + bi$ stored as separate real and imaginary parts.

Question 112 HARD

Calculate $e^{i\theta}$ using Euler's formula and verify $e^{i\pi} + 1 = 0$. Compute powers of complex numbers using De Moivre's theorem.

Question 113 EXTREME

Find all nth roots of a complex number using polar form: convert to polar, apply root formula, convert back to rectangular form.

Question 114

Evaluate complex polynomial $P(z) = z^4 - 1$ at 10 points around unit circle and analyze the pattern.

Question 115 HARD

Implement complex logarithm: $\ln(z) = \ln|z| + i\arg(z)$ handling branch cuts properly with conditionals.

13 Fourier Analysis & Signal Processing

Question 116 EXTREME

Calculate Discrete Fourier Transform (DFT) of 8-point real sequence using direct formula: $X[k] = \sum_{n=0}^{7} x[n]e^{-i2\pi kn/8}$.

Question 117 HARD

Implement the Cooley-Tukey FFT algorithm for 8 points using butterfly operations with loops (no recursion).

Question 118

Compute the convolution of two sequences of length 5 each: $y[n] = \sum_k x[k]h[n-k]$ using direct method.

Question 119 HARD

Calculate Fourier series coefficients for square wave: $f(x) = \begin{cases} 1 & 0 \leq x < \pi \\ -1 & \pi \leq x < 2\pi \end{cases}$ up to 20 harmonics.

Question 120 EXTREME

Implement autocorrelation function for a sequence and identify periodicity using peak detection with loops.

14 Probability & Statistics (Computational)

Question 121

Implement a Linear Congruential Generator (LCG) for pseudo-random numbers: $X_{n+1} = (aX_n + c) \mod m$. Generate 1000 numbers and compute mean/variance.

Question 122 HARD

Calculate sample mean, variance, standard deviation, skewness, and kurtosis for a dataset of 20 numbers stored as separate variables.

Question 123

Compute the median and quartiles (Q1, Q3) of a dataset by implementing selection algorithm using loops (no sorting allowed).

Question 124 EXTREME

Implement Box-Muller transform to generate normally distributed random numbers from uniform random numbers using LCG.

Question 125 HARD

Calculate Pearson correlation coefficient between two sequences of 15 values each. Determine if correlation is significant.

Question 126

Perform linear regression on 10 data points: find slope and intercept of best-fit line using least squares method.

Question 127 EXTREME

Implement the bootstrap method: resample 100 times from original 20 data points to estimate confidence interval for the mean.

Question 128 HARD

Calculate the chi-square statistic for goodness-of-fit test with 5 categories. Determine degrees of freedom and interpret result.

Question 129

Compute moving averages (window size 5) and exponential moving average (=0.3) for a time series of 30 values.

Question 130 HARD

Implement the Kolmogorov-Smirnov test statistic to compare an empirical distribution (20 samples) with uniform distribution.

15 Graph Theory Algorithms

Question 131 EXTREME

Implement depth-first search (DFS) on a graph represented as 6×6 adjacency matrix using loop-based stack simulation.

Question 132 HARD

Perform breadth-first search (BFS) on a 6-node graph to find shortest path lengths from source to all other nodes.

Question 133

Detect if a directed graph (5×5 adjacency matrix) contains a cycle using DFS-based approach with state tracking.

Question 134 EXTREME

Find strongly connected components of a directed graph using Kosaraju's algorithm with two DFS passes.

Question 135 HARD

Implement Prim's algorithm to find minimum spanning tree of weighted undirected graph (6 nodes) using loops.

Question 136

Use Kruskal's algorithm to find MST by sorting edges by weight (implement sorting) and checking for cycles.

Question 137 EXTREME

Solve the traveling salesman problem for 6 cities using branch and bound approach with loops (find approximate solution).

Question 138 HARD

Implement topological sorting of a directed acyclic graph (DAG) with 7 nodes using Kahn's algorithm.

Question 139

Find the maximum flow in a flow network using Ford-Fulkerson algorithm with BFS (Edmonds-Karp) for 6 nodes.

Question 140 EXTREME

Implement Bellman-Ford algorithm to detect negative cycles and find shortest paths in weighted directed graph.

16 Computational Geometry

Question 141 HARD

Implement Graham scan algorithm to find convex hull of 10 points in 2D plane using loops for sorting and stack simulation.

Question 142

Determine if two line segments intersect using cross product method. Handle all special cases (parallel, collinear, overlapping).

Question 143 EXTREME

Find the closest pair of points among 15 2D points using divide-and-conquer approach simulated with loops.

Question 144 HARD

Compute the area of a polygon (up to 12 vertices) using the Shoelace formula with loops.

Question 145

Implement point-in-polygon test using winding number algorithm for complex polygons with loops.

17 Final Challenge Problems

Question 146 EXTREME

Implement the Fast Inverse Square Root algorithm (Quake III) $\frac{1}{\sqrt{x}}$ using bit manipulation simulation and Newton-Raphson refinement.

Question 147 EXTREME

Solve the N-Queens problem for N=8 using backtracking simulated with loops and state variables. Count all distinct solutions.

Question 148 EXTREME

Implement the Aho-Corasick algorithm for multiple pattern matching: search for 5 different patterns in a text string (100 chars) using finite automaton built with loops.

Question 149 EXTREME

Simulate a cellular automaton (Conway's Game of Life) on a 10×10 grid for 50 generations. Track population evolution.

Question 150 EXTREME

Implement the Shor-Wagstaff algorithm for finding the period of modular exponentiation: find smallest r where $a^r \equiv 1 \pmod{N}$ and use it for integer factorization attempt.

18 Grading Rubric

Assessment Criteria

- **Correctness (40%):** Does the solution produce correct results for all test cases?
- **Efficiency (20%):** Is the algorithm optimized within constraint limitations?
- **Code Quality (20%):** Clear logic, proper variable names, edge case handling?
- **Documentation (10%):** Well-commented code explaining approach and logic?
- **Testing (10%):** Multiple test cases provided with expected outputs?

19 Submission Guidelines

Important Instructions

1. Submit solutions in a single well-organized document
2. Each solution must include:
   - Problem number and title
   - Algorithm explanation (pseudocode or description)
   - Complete code with comments
   - Test cases and outputs
   - Time complexity analysis
3. Use proper formatting and indentation
4. Deadline: **As announced by instructor**
5. Late submissions: -10% per day

"In theory, theory and practice are the same. In practice, they are not."

– Albert Einstein

Master these fundamentals - they form the foundation of all complex algorithms.

Instructor: Thomas Basyal
Department: AI/ML Engineering
Email: thomas.basyal@university.edu
Office Hours: TBD