

Workshop Challenge: Set B (Advanced)

Topic: Nested Loops & Computational Logic Difficulty: Intermediate to Advanced

Part 1: Advanced Visual Patterns

Goal: Complex index manipulation.

1. The Diamond Shape Write a program to print a diamond made of stars.

Input: n=3 (half-height)

Output:

Plaintext

```
*  
* *  
* * *  
* *  
*
```

2. The "Hourglass" Print an hourglass pattern of numbers.

Input: 5

Output:

Plaintext

```
1 2 3 4 5  
2 3 4  
3  
2 3 4  
1 2 3 4 5
```

3. The Binary Right Angle Print a triangle where rows alternate between starting with 1 and 0.

Output:

Plaintext

```
1  
0 1  
1 0 1  
0 1 0 1
```

4. The Box with a Cross Print a square box where the border and the diagonals are stars (*), and the rest are spaces.

Hint: Print * if row==0, row==end, col==0, col==end, row==col, or row+col==end.

5. The Alphabet Pyramid Print letters instead of numbers.

Output:

Plaintext

A

B B

C C C

D D D D

Hint: Use chr(65 + i) in Python to generate 'A', 'B', etc.

Part 2: Data & String Processing

Goal: Using nested loops for string and list analysis.

6. Longest Common Prefix Given a list of strings ["apple", "ape", "april"], find the longest starting string that matches all of them (result: "ap").

Logic: The outer loop iterates through the characters of the first word; the inner loop checks that character against all other words.

7. Find Duplicate Characters Take a string input and print any character that appears more than once.

Input: "programming"

Output: r, g, m

8. Is it a Palindrome? (Manual Check) Check if a string is a palindrome without using the slice method [::-1].

Logic: Use a loop to compare index with len - index - 1.

9. Matrix Multiplication (Dot Product) Unlike matrix addition, multiply the Row of Matrix A by the Column of Matrix B. (Classic Linear Algebra).

Requirement: 3 Nested Loops (Rows of A, Cols of B, Common Dimension).

10. Transpose and Rotate Rotate a 3x3 matrix 90 degrees clockwise.

Input: [[1, 2], [3, 4]]

Output: [[3, 1], [4, 2]]

Part 3: Number Theory & Math

Goal: Mathematical brute-forcing.

11. Pythagorean Triplets Find all sets of numbers (a, b, c) up to 20 such that $a^2 + b^2 = c^2$.

Output: (3, 4, 5), (5, 12, 13)...

12. Armstrong Numbers Find all numbers between 100 and 999 where the sum of the cubes of the digits equals the number itself.

Example: 153 $\rightarrow 1^3 + 5^3 + 3^3 = 153$.

13. Perfect Numbers Find numbers where the sum of divisors equals the number.

Example: 6 $\rightarrow 1 + 2 + 3 = 6$.

Range: Find all between 1 and 100.

14. Prime Factorization Take a number n and print all its prime factors.

Input: 12

Output: 2, 2, 3

15. Pascal's Triangle Print the first 5 rows of Pascal's Triangle (where each number is the sum of the two above it).

Output:

Plaintext

1

1 1

1 2 1

1 3 3 1

Part 4: Real-World Simulations

Goal: Simulating time and space.

16. The Digital Clock Print every minute of a day in HH:MM format.

Output: 00:00, 00:01 ... 23:59.

Loops: Hour (0-23), Minute (0-59).

17. The Odometer Simulate a 3-digit counter that skips the digit 4 (superstitious counter).

Output: 000, 001, 002, 003, 005...

18. Coordinate Distance Calculator Given a list of points [(1,1), (2,2), (5,5)], calculate the distance between every pair of points using the distance formula.

19. Image Blur (Average Filter) Given a 3x3 grid of numbers (pixels), create a new grid where every cell is the average of itself and its neighbors. (Simplify to just row-neighbors if needed).

20. The "Snake" Grid Print numbers 1 to 9 in a 3x3 grid, but snake the direction (left-to-right, then right-to-left).

Output:

Plaintext

1 2 3

6 5 4

7 8 9