🐍 PYTHON MASTERY PRACTICE (DETAILED & LOGIC-HEAVY)

📘 MODULE 1: Python Basics, Variables & Input

🔹 Core Logic Questions

1. Write a program that takes **name, age, and marks** of a student and prints:
    a. eligibility for voting
    b. grade (A/B/C)
    c. year of birth

(Use type casting + conditions)

2. Take a number and:
    a. check its data type
    b. convert it to all possible numeric types
    c. explain data loss if any
3. Write a program that:
    a. swaps two numbers using **3 different methods**
    b. explain which is best and why
4. Accept input in a single line:

```
10 20 30 40
```

Convert it into a list of integers and print sum and average.

5. Write a program that demonstrates:
    a. Python is dynamically typed
    b. variables can change type at runtime

📓 MODULE 2: Operators & Control Flow (LOGIC BUILDING)

◇ Decision Making Mastery

1. Create a **menu-driven calculator** using `if-elif`.
2. Write a program to check:
    a. leap year
    b. century year
    c. edge cases (1900, 2000, 2100)
3. Given 3 sides of triangle:
    a. validate triangle
    b. classify as equilateral, isosceles, scalene

◇ Loop Power Questions

4. Print all numbers between 1–500 that are:
    a. divisible by 7
    b. not divisible by 5
5. Write a program to reverse a number **without converting to string**.
6. Find:
    a. sum of digits
    b. product of digits
    c. count of digits
    (single loop)
7. Generate **Fibonacci series** up to N terms using loop.
8. Print all **Armstrong numbers** between 1 and 1000.

📝 MODULE 3: String Manipulation (VERY IMPORTANT)

◇ String Logic Questions

1. Check whether two strings are **anagrams**.
2. Find **first non-repeating character** in a string.
3. Count frequency of each word in a paragraph.
4. Reverse **each word** in a sentence.
5. Compress a string:

```
aaabbc → a3b2c1
```

◇ Advanced String Thinking

6. Validate password:
    a. min 8 chars
    b. at least 1 digit
    c. 1 uppercase
    d. 1 special character
7. Remove all duplicate characters **without using set**.
8. Find longest word in a sentence.
9. Check palindrome **ignoring spaces & cases**.

📱 MODULE 4: Data Structures (CORE PYTHON POWER)

◇ Lists

1. Rotate list by k positions.
2. Find second largest element **without sorting**.
3. Merge two sorted lists into one sorted list.
4. Find common elements between two lists **without using set**.

◇ Tuples

5. Convert tuple of tuples into dictionary.
6. Find frequency of elements in tuple.

◇ Dictionaries (VERY IMPORTANT)

7. Word frequency counter using dictionary.
8. Find key with maximum value.
9. Invert dictionary:

```
{'a':1,'b':2} → {1:'a',2:'b'}
```

◇ Sets

10. Remove duplicates from list using set logic.
11. Check if two lists have common elements.

📓 MODULE 5: Functions & RECURSION (LOGIC BOOSTER)

◇ Functions

1. Write a function to check prime number.
2. Write a function to generate first N primes.
3. Write function returning:
    a. min
    b. max
    c. sum

(single function)

🔁 RECURSION (MUST MASTER)

4. Factorial using recursion (trace call stack).
5. Fibonacci using recursion.
6. Power of number (x^n).
7. Sum of digits using recursion.
8. Reverse number using recursion.
9. Count digits using recursion.
10. Check palindrome using recursion.

📘 MODULE 6: Object-Oriented Programming (REAL WORLD)

◇ Design Thinking

1. Create a `BankAccount` class:
    a. deposit
    b. withdraw
    c. balance check
2. Demonstrate inheritance using:
    a. Person → Employee
3. Override method in child class.
4. Use encapsulation to protect balance variable.
5. Show polymorphism using same method name.

6. Create class with class variable and instance variable.

🧱 MODULE 7: Advanced Python (INDUSTRY READY)

◇ Exception Handling

1. Handle divide by zero.
2. Validate age input with custom exception.
3. Handle file not found error.

◇ File Handling

4. Read a file and count:
    a. lines
    b. words
    c. characters
5. Copy content from one file to another.

◇ JSON & Pickle

6. Store student data in JSON.
7. Serialize and deserialize object using pickle.

◇ Regular Expressions

8. Validate email.
9. Validate Indian mobile number.
10. Extract all numbers from string.

🧠 HOW THIS MAKES YOU A MASTER

If you can solve **90% of these without help**:

✅ Python fundamentals = strong

✅ Logic = strong

✅ Interview ready

- [x] Competitive coding ready