

## Python Practice Problems

### 1. Basic Python Syntax and Operations

#### Level 1: Easy

1. Write a program to add two numbers entered by the user.
2. Calculate the sum of all integers from 1 to a user-specified number.
3. Find the area of a circle given the radius (user input).
4. Print all even numbers between 1 and 100.
5. Given two numbers, return their maximum.

#### Level 2: Intermediate

1. Write a function to check if a given year is a leap year.
2. Create a function to calculate the nth Fibonacci number.
3. Implement a basic calculator that performs addition, subtraction, multiplication, and division based on user input.
4. Generate a multiplication table for a number provided by the user.
5. Write a program to reverse a given integer.

#### Level 3: Advanced

1. Create a function to calculate the factorial of a number using recursion.
  2. Implement a function that checks whether a string is a palindrome.
  3. Write a program to find the sum of all multiples of 3 and 5 below a user-specified number.
  4. Create a function to find the greatest common divisor (GCD) of two numbers.
  5. Implement a number guessing game where the computer selects a random number between 1 and 100, and the user has to guess it.
- 

### 2. Number Theory and Basic Math Problems

#### Level 1: Easy

1. Write a program to check if a number is prime.
2. List all prime numbers up to a given limit.
3. Find the sum of the digits of a user-inputted number.
4. Check if a number is even or odd.

5. Write a program to count the number of factors a number has.

#### Level 2: Intermediate

1. Write a function to calculate the LCM of two numbers.
2. Create a program to find all pairs of numbers in a list that add up to a specified sum.
3. Generate the first  $n$  prime numbers (where  $n$  is provided by the user).
4. Find the  $n$ th term of an arithmetic sequence given the first term and common difference.
5. Implement a function to generate a list of all perfect squares less than a given number.

#### Level 3: Advanced

1. Create a function that returns all prime factors of a given number.
  2. Implement a program to generate the Collatz sequence for a given number.
  3. Find all Pythagorean triplets  $(a, b, c)$  such that  $a^2 + b^2 = c^2$  and all numbers are less than 100.
  4. Create a program to find the sum of all divisors of a given number.
  5. Write a function that returns the  $n$ th term of the Fibonacci sequence using dynamic programming.
- 

### 3. Data Structures and Algorithms

#### Level 1: Easy

1. Write a program to find the maximum and minimum elements in a list.
2. Create a list of numbers and print their squares using a list comprehension.
3. Count the number of occurrences of a specific element in a list.
4. Given a list of numbers, find their average.
5. Write a program to merge two lists and sort the result.

#### Level 2: Intermediate

1. Implement a function to find the intersection of two lists.
2. Given a string, count the occurrences of each character using a dictionary.
3. Write a program to flatten a nested list (e.g., `1,2,[3,4],51, 2, [3, 4], 51,2,[3,4],5` becomes `1,2,3,4,51, 2, 3, 4, 51,2,3,4,5`).
4. Implement a function to remove duplicates from a list.

5. Write a program to rotate a list to the right by  $n$  places.

#### Level 3: Advanced

1. Implement a binary search on a sorted list.
  2. Create a program to check if a list is sorted in ascending order.
  3. Write a function that returns the longest increasing subsequence of a list of numbers.
  4. Create a function to calculate the Cartesian product of two lists.
  5. Implement a function that returns all possible subsets of a list (the power set).
- 

### 4. Mathematical Problem-Solving and Logic

#### Level 1: Easy

1. Write a program to find the sum of the first  $n$  natural numbers.
2. Calculate the sum of an arithmetic series given its first term, common difference, and number of terms.
3. Check if a number is an Armstrong number (e.g., 153 is an Armstrong number).
4. Print the first  $n$  terms of the sequence 2, 4, 8, 16, ...
5. Write a function to find the square root of a number (using simple approximation).

#### Level 2: Intermediate

1. Write a function to find the sum of the first  $n$  terms of a geometric series.
2. Implement a program to solve quadratic equations given coefficients  $a$ ,  $b$ , and  $c$ .
3. Write a program to determine if a number is a perfect number (equal to the sum of its proper divisors).
4. Create a function to calculate the sum of the first  $n$  odd numbers.
5. Write a program to find all the numbers in a list that are palindromic.

#### Level 3: Advanced

1. Implement a program to find the  $n$ th Catalan number.
2. Write a program to generate Pascal's triangle up to a given number of rows.
3. Create a function that generates all possible permutations of a string.
4. Write a program to evaluate a mathematical expression given as a string (e.g., " $3 + 5 * 2$ ").
5. Implement a function that returns the  $n$ th Lucas number.

---

## 5. Statistics and Probability Fundamentals

### Level 1: Easy

1. Write a program to calculate the mean of a list of numbers.
2. Find the median of a list of numbers.
3. Count the mode(s) of a list of numbers.
4. Generate 10 random integers between 1 and 100 and print them.
5. Calculate the range of a list of numbers (difference between the maximum and minimum).

### Level 2: Intermediate

1. Write a program to calculate the variance of a list of numbers.
2. Create a function to simulate rolling two six-sided dice 1000 times and display the frequency of each sum.
3. Calculate the probability of drawing a specific card from a deck of cards.
4. Implement a function to normalize a list of numbers (rescale them to a range of 0 to 1).
5. Generate a random sample of 10 elements from a list.

### Level 3: Advanced

1. Write a function to calculate the standard deviation of a list of numbers.
  2. Implement a function to calculate the cumulative distribution function (CDF) for a normal distribution given mean and standard deviation.
  3. Simulate flipping a coin  $n$  times and count the longest run of heads or tails.
  4. Write a program to calculate correlation between two lists of numbers.
  5. Implement a program to simulate the Monty Hall problem and calculate the probability of winning if the player switches doors.
- 

## 6. NumPy Basics and Vectorized Operations

### Level 1: Easy

1. Create a NumPy array with values from 0 to 9.
2. Generate a 3x3 identity matrix using NumPy.
3. Create a NumPy array with random values between 0 and 1.

4. Write a program to reshape a 1D NumPy array into a 2D array.
5. Find the sum of all elements in a NumPy array.

#### Level 2: Intermediate

1. Create a NumPy array of even numbers between 10 and 50.
2. Write a program to multiply two NumPy arrays element-wise.
3. Generate a 2x3 matrix with normally distributed random values.
4. Implement a program to transpose a NumPy array.
5. Write a function to find the dot product of two 1D NumPy arrays.

#### Level 3: Advanced

1. Create a function to perform matrix multiplication on two 2D NumPy arrays.
  2. Implement a function that normalizes a NumPy array (subtract mean, divide by std deviation).
  3. Write a program to calculate the determinant of a 3x3 matrix.
  4. Create a function to generate a NumPy array of sine values given an array of angles in radians.
  5. Implement a program to perform element-wise comparison between two NumPy arrays and return a Boolean array.
- 

## 7. Data Visualization with Matplotlib

#### Level 1: Easy

1. Plot a line graph of  $y = x^2$  for  $x$  values from -10 to 10.
2. Generate a bar chart showing the population of five cities.
3. Create a scatter plot of 20 random  $(x, y)$  pairs.
4. Draw a histogram of 100 random numbers between 1 and 100.
5. Plot a sine wave between 0 and  $2\pi$ .

#### Level 2: Intermediate

1. Create a subplot with 2 graphs: one showing  $y = x^2$  and the other showing  $y = x^3$ .
2. Plot the relationship between a list of years and corresponding rainfall amounts as a line plot.
3. Create a pie chart of market share for four companies.

4. Plot the functions  $\sin(x)$  and  $\cos(x)$  on the same graph with different colors and labels.
5. Customize a plot by adding titles, axis labels, and a grid.

### Level 3: Advanced

1. Generate a heatmap of a 5x5 matrix of random values.
2. Create a scatter plot with different sizes and colors for each point.
3. Plot a bar chart with error bars representing standard deviations of data.
4. Animate a plot showing the growth of a population over time.
5. Create a polar plot of a function like  $r = \sin(\theta)$ .