## Problem 1: "Admission Eligibility"

*Story*: Rajesh is an admission officer at a prestigious university. He needs a program to determine whether a student is eligible for admission based on their test scores. The eligibility criteria are as follows:

* + The student must score above 75 in Math and above 80 in Science.
  + The average score in all subjects must be at least 70.

Write a Python program that takes input for Math, Science, and English scores of a student and prints whether the student is eligible for admission or not.

## Problem 2: "Temperature Converter"

*Story*: Priya is a weather analyst who needs a program to convert temperatures from Celsius to Fahrenheit and vice versa. She often needs to perform these conversions quickly during her analysis.

Write a Python program that asks the user to input a temperature along with its unit (C for Celsius or F for Fahrenheit) and then converts it to the other unit.

## Problem 3: "Guess the Number Game"

*Story*: Anand and his friends are playing a number guessing game. The program generates a random number between 1 and 100, and the player has to guess the number. The program provides hints such as "too high" or "too low" after each guess until the player guesses the correct number.

Write a Python program that generates a random number and lets the user guess until they guess correctly. Provide appropriate feedback after each guess.

## Problem 4: "Grade Calculator"

*Story*: Meena is a teacher who needs a program to calculate the grades of her students. She follows a simple grading system where:

* + 90 and above: A
  + 80-89: B
  + 70-79: C
  + 60-69: D
  + Below 60: F

Write a Python program that takes input for a student's score and prints their corresponding grade.

## Problem 5: "Simple Interest Calculator"

*Story*: Rohit is studying finance and needs a program to calculate simple interest for his assignments. He wants to quickly calculate the interest based on the principal amount, rate, and time.

Write a Python program that takes input for principal amount, rate of interest, and time period, and then calculates and prints the simple interest.

## Problem 6: "Vowel Counter"

*Story*: Ritu is analyzing some text data and wants to know the frequency of each vowel (a, e, i, o, u) in the text. She needs a program to count the occurrences of each vowel.

Write a Python program that takes a string input from the user and counts the occurrences of each vowel. Ignore case sensitivity.

## Problem 7: "Palindrome Checker"

*Story*: Sanjay is learning about palindromes in his computer science class. He wants to check if a given word is a palindrome or not. A palindrome is a word that reads the same forwards and backwards.

Write a Python program that takes a word as input and checks whether it is a palindrome or not.

## Problem 8: "Leap Year Checker"

*Story*: Nisha is a calendar planner who wants to check if a given year is a leap year or not. She needs a program to quickly determine if a year is divisible by 4 but not by 100 unless it is also divisible by 400.

Write a Python program that takes a year as input and checks if it is a leap year or not.

## Problem 9: "Factorial Calculator"

*Story*: Suresh is studying permutations and combinations and needs a program to calculate factorials for his assignments. He wants a program to quickly calculate the factorial of a given number.

Write a Python program that takes an integer input from the user and calculates its factorial.

## Problem 10: "FizzBuzz Game"

*Story*: Deepak is teaching programming to elementary school students and wants to play the FizzBuzz game with them. In this game, players count numbers but replace numbers divisible by 3 with "Fizz", numbers divisible by 5 with "Buzz", and numbers divisible by both 3 and 5 with "FizzBuzz".

Write a Python program that prints the numbers from 1 to 100, but replaces multiples of 3 with "Fizz", multiples of 5 with "Buzz", and multiples of both 3 and 5 with "FizzBuzz".

## Problem 11: "Unique Elements"

*Story*: Ananya is analyzing data and needs to find unique elements in a list. She wants a program to quickly identify and display unique elements in a given list.

Write a Python program that takes a list as input and prints the unique elements present in the list.

## Problem 12: "List Manipulation"

*Story*: Rahul is managing a list of tasks for his project. He needs a program to perform various operations on the list such as adding, removing, and updating tasks.

Write a Python program that allows the user to perform the following operations on a list of tasks:

* + Add a task
  + Remove a task
  + Update a task
  + Display all tasks

## Problem 13: "Set Operations"

*Story*: Kavya is studying sets and wants to perform various set operations. She needs a program to demonstrate set operations like union, intersection, and difference.

Write a Python program that takes two sets as input and performs the following set operations:

* + Union
  + Intersection
  + Difference (both A - B and B - A)
  + Symmetric difference

## Problem 14: "Dictionary Manipulation"

*Story*: Arjun is managing a dictionary of employee records and needs a program to perform operations like adding new employees, updating their details, and retrieving information.

Write a Python program that allows the user to perform the following operations on a dictionary of employee records:

* + Add a new employee
  + Update employee details
  + Retrieve employee information
  + Display all employees

## Problem 15: "List Comprehensions"

*Story*: Naveen is learning about list comprehensions and wants to practice creating lists using this technique. He needs a program to generate lists based on specific criteria.

Write Python programs using list comprehensions to accomplish the following tasks:

* + Generate a list of even numbers between 1 and 20.
  + Generate a list of squares of numbers from 1 to 10.
  + Generate a list of names with more than 5 characters from a given list of names.

## Problem 16: "Set Comprehensions"

*Story*: Priya is intrigued by set comprehensions and wants to explore creating sets using this technique. She needs a program to generate sets based on specific criteria.

Write Python programs using set comprehensions to accomplish the following tasks:

* + Generate a set of vowels from a given string.
  + Generate a set of unique characters from a given string.
  + Generate a set of prime numbers between 1 and 50.

## Problem 17: "Dictionary Comprehensions"

*Story*: Rohit is fascinated by dictionary comprehensions and wants to practice creating dictionaries using this technique. He needs a program to generate dictionaries based on specific criteria.

Write Python programs using dictionary comprehensions to accomplish the following tasks:

* + Generate a dictionary mapping numbers to their squares from 1 to 10.
  + Generate a dictionary mapping names to their lengths from a given list of names.
  + Generate a dictionary mapping vowels to their counts in a given string.

## Problem 18: "Matrix Operations"

*Story*: Sneha is studying matrix operations and wants to perform various operations such as addition, multiplication, and transpose. She needs a program to demonstrate these operations.

Write a Python program that allows the user to perform the following matrix operations:

* + Matrix addition
  + Matrix multiplication
  + Transpose of a matrix

## Problem 19: "Tuple Operations"

*Story*: Rajesh is managing employee records and wants to use tuples to store information such as employee ID, name, and salary. He needs a program to perform operations like adding new records and displaying all records.

Write a Python program that allows the user to perform the following operations on a list of employee records stored as tuples:

* + Add a new employee record
  + Update employee details
  + Retrieve employee information
  + Display all employee records

## Problem 20: "Frequency Count"

*Story*: Anjali is analyzing text data and wants to count the frequency of each word in a given text. She needs a program to quickly generate a dictionary mapping words to their frequencies.

Write a Python program that takes a string input from the user and generates a dictionary mapping words to their frequencies.

## Problem 21: "Virtual Environment Manager"

Story: Priya is working on multiple Python projects simultaneously and wants to manage their dependencies effectively. She needs a program to create and manage virtual environments for each project.

Write a Python program that allows Priya to create, activate, and deactivate virtual environments for her projects.

## Problem 22: "Function Docstring Generator"

Story: Rahul is writing a Python library and wants to ensure that all functions are properly documented. He needs a program that automatically generates docstrings for his functions based on their parameters and return values.

Write a Python program that prompts the user to input function parameters and return values, and generates a docstring template for the function.

## Problem 23: "Function Argument Handler"

Story: Anjali is working on a project that involves handling various types of function arguments. She needs a program to handle positional arguments, named arguments, and variable arguments effectively.

Write a Python program that demonstrates the use of positional arguments, named arguments, and variable arguments in function calls.

## Problem 24: " Method Overloading with multipledispatch"

Story: Aditi is learning about method overloading in Python and wants to write a program to demonstrate method overloading using the multipledispatch decorator module. She wants to define multiple functions with the same name but different parameter types and have the appropriate function called based on the parameters passed.

Write a Python program that uses the decorator ‘multipledispatch’ to overload variations of ‘add’ function.

## Problem 25: "Special Functions Explorer"

Story: Meena wants to explore special functions in Python such as generators, iterators, and lambda functions. She needs a program that demonstrates the use of these special functions.

Write a Python program that showcases the usage of generators, iterators, and lambda functions in various scenarios.

## Problem 26: "Utility Function Showcase"

Story: Rohit is learning about utility functions such as map, filter, and reduce in Python. He wants to understand their usage in real-world scenarios.

Write a Python program that demonstrates the usage of map, filter, and reduce functions to perform common data manipulation tasks.

## Problem 27: "Module Importer and User"

Story: Ritu is working on a project that involves using external modules. She wants to understand how to import modules and use their functionalities effectively.

Write a Python program that demonstrates importing custom and built-in modules and utilizes their functionalities in the main program.

## **P**roblem 28: "Calculator Application"

Story: Suresh is working on a calculator application and needs a program to perform basic arithmetic operations such as addition, subtraction, multiplication, and division. He wants to ensure that the calculator is user-friendly and provides accurate results.

Write a Python program that simulates a calculator application, allowing users to perform basic arithmetic operations on two numbers.

## Problem 29: "Fibonacci Sequence Generator"

Story: Deepak is studying sequences and series and wants to generate the Fibonacci sequence up to a specified limit. He needs a program to generate the Fibonacci sequence efficiently.

Write a Python program that generates the Fibonacci sequence up to a specified limit provided by the user.

## Problem 30: "Random Number Generator"

Story: Anand is working on a project that requires generating random numbers for simulations. He needs a program to generate random numbers within a specified range.

Write a Python program that generates a specified number of random integers within a given range and prints them.

## Problem 31: "Email Validator"

Story: Rajesh is developing a registration system for a website and needs to validate email addresses entered by users. He wants to ensure that the email addresses follow the correct format.

Write a Python program that takes an email address as input and validates whether it follows the correct email format using regular expressions.

## Problem 32: "Phone Number Extractor"

Story: Priya is working on a project that involves processing text messages and needs to extract phone numbers from the messages. She wants to extract phone numbers regardless of their formatting.

Write a Python program that extracts phone numbers from a given text message using regular expressions.

Sample texts:

1. "Hello, my phone number is 123-456-7890. Please call me back."
2. "You can reach me at 987.654.3210 or via email at example@example.com."
3. "Contact us at (555) 123-4567 for more information."
4. "Our office number is 1112223333. Feel free to give us a call."
5. "For urgent inquiries, please dial 1-800-555-1234."

## Problem 33: "URL Extractor"

Story: Anjali is analyzing a web page and needs to extract URLs from the page content. She wants to extract both HTTP and HTTPS URLs.

Write a Python program that extracts URLs from a given text using regular expressions, considering both HTTP and HTTPS URLs.

Sample text:

* "Welcome to our website. You can visit us at https://www.example.com for more information. Our online store is available at http://store.example.com. Feel free to explore our products and services. Contact us via email at contact@example.com for any queries."

## Problem 34: "Word Finder"

Story: Sanjay is working on a text analysis project and needs to find occurrences of a specific word in a given text. He wants to find all occurrences of the word regardless of its case.

Write a Python program that finds all occurrences of a specific word in a given text, ignoring case sensitivity, using regular expressions.

## Problem 35: "Password Strength Checker"

Story: Meena is developing a password strength checker for a website and wants to ensure that passwords meet certain criteria. She wants to check if a password contains at least one uppercase letter, one lowercase letter, one digit, and one special character.

Write a Python program that checks the strength of a password based on certain criteria using regular expressions.

## Problem 36: "IP Address Validator"

Story: Rohit is building a network monitoring tool and needs to validate IP addresses entered by users. He wants to ensure that the IP addresses follow the correct format.

Write a Python program that takes an IP address as input and validates whether it follows the correct IP address format using regular expressions.

The regular expression pattern should match an IP address in the form of "X.X.X.X" where X is a number between 0 and 255. The conditions applied to check the validity of the IP address should be as follows:

* Each segment of the IP address should be between 0 and 255.
* Each segment should not contain leading zeros, except for the value 0 itself.
* There should be exactly 3 dots separating the 4 segments.

If the IP address matches this pattern, it is considered valid; otherwise, it is invalid.

## Problem 37: "Date Extractor"

Story: Ritu is working on a project that involves processing text documents containing dates. She needs to extract dates from the documents in various formats (e.g., DD-MM-YYYY, MM/DD/YYYY).

Write a Python program that extracts dates from a given text document in different formats using regular expressions.

## Problem 38: "Word Counter"

Story: Suresh is analyzing a large text document and wants to count the occurrences of each word. He wants to ignore punctuation and consider words regardless of their case.

Write a Python program that counts the occurrences of each word in a given text document, ignoring punctuation and case sensitivity, using regular expressions.

## Problem 39: "HTML Tag Remover"

Story: Deepak is working on a web scraping project and needs to remove HTML tags from the extracted content. He wants to extract only the text content from HTML documents.

Write a Python program that removes HTML tags from a given HTML document and extracts the text content using regular expressions.

## Problem 40: "License Plate Extractor"

Story: Anand is developing an automatic license plate recognition system and needs to extract license plate numbers from images of vehicles. He wants to extract license plate numbers regardless of their formatting.

Write a Python program that extracts license plate numbers from images of vehicles using regular expressions.