**Python Application Challenges (Non-MCQ)**

**Data Structures and Logic**

1. **Inventory Stock Check (Dictionary & If/Else):**
   * **Task:** Define a dictionary named inventory with at least three items and their stock levels (e.g., {'pen': 50, 'paper': 100, 'stapler': 5}).
   * **Logic:** Write a function check\_stock(item\_name) that prints "In Stock" if the stock level is greater than 10, "Low Stock" if it's 1 to 10, and "Out of Stock" if it's 0. If the item is not in the dictionary, print "Item not found."
2. **Unique Words Counter (Set & String):**
   * **Task:** Given a sentence, write a program to count the number of **unique words** (case-insensitive).
   * **Hint:** Convert the sentence to lowercase, split it into words, and then use a Set to handle uniqueness.
3. **Tuple List Filtering (Tuple & List & Loop):**
   * **Task:** Given a list of tuples, where each tuple represents a product and its price (e.g., products = [('A', 15), ('B', 5), ('C', 22), ('D', 18)]).
   * **Logic:** Write a loop to create a **new list** containing only the names of products whose price is greater than 15.
4. **Dictionary Key Reversal (Dictionary & Loop):**
   * **Task:** Given a dictionary (e.g., data = {'a': 1, 'b': 2}), write a program to create a **new dictionary** where the original keys and values are swapped (i.e., the values become the keys, and the keys become the values).
   * **Constraint:** Assume all original values are unique and hashable.
5. **List De-duplication (Set & List):**
   * **Task:** Given a list with duplicate entries (e.g., my\_list = [10, 20, 10, 30, 20, 40]), write the shortest possible code to remove all duplicates and store the resulting unique values back into a **new list** (order doesn't matter).

**Conditional and Iteration Logic**

1. **Grade Calculator (If/Elif/Else):**
   * **Task:** Write a program that takes a numerical score as input (between 0 and 100) and prints the corresponding letter grade.
   * **Grading:** A (90+), B (80−89), C (70−79), D (60−69), F (below 60).
2. **Odd/Even Counter (Loop & Modulus Operator):**
   * **Task:** Given a list of integers (e.g., nums = [1, 4, 7, 10, 13, 16]).
   * **Logic:** Write a for loop that iterates through the list and counts how many numbers are **even** and how many are **odd**. Print the final counts.
3. **Vowel Counter (String & Set & Loop):**
   * **Task:** Write a function that takes a word as a string input.
   * **Logic:** Use a Set containing all lowercase vowels ('a', 'e', 'i', 'o', 'u') and count how many vowels are in the input word (case-insensitive).
4. **Basic Menu System (Input & If/Elif/Else):**
   * **Task:** Create a simple text-based menu that displays three options (e.g., 1: Start Game, 2: Load Game, 3: Exit).
   * **Logic:** Prompt the user for a choice. Use an if/elif/else structure to print a different message for each valid choice, and a "Invalid choice" message for any other input.
5. **Tuple of Squares (Comprehension):**
   * **Task:** Use a **list comprehension** to generate a list of the squares of numbers from 1 to 5.
   * **Final Step:** Convert that resulting list into an **immutable tuple** and print it.