Python Practice Problems – Level 3

### ****Logic & Number Problems****

1. **Sum of Unique Numbers**  
   sum\_unique(lst) – Return the sum of numbers that appear only once in the list.
2. **Count Digits in a Number**  
   count\_digits(n) – Return how many digits are in a number (e.g., 752 → 3).
3. **Find Missing Number**  
   find\_missing(lst) – Given a list from 1 to n with one number missing, return the missing one.
4. **Generate Multiples**  
   generate\_multiples(n, count) – Return the first count multiples of n.
5. **Convert Time to Minutes**  
   to\_minutes(hours, minutes) – Return total time in minutes.
6. **Sum of Digits Until One Digit**  
   reduce\_to\_digit(n) – Keep summing digits until one digit remains. (e.g., 9875 → 2)
7. **Check for Balanced Parentheses**  
   is\_balanced(s) – Return True if every ( has a matching ) in the string.

### 🧠 ****Word & String Challenges****

1. **Check Anagram**  
   is\_anagram(w1, w2) – Return True if words are anagrams (e.g., "listen" and "silent").
2. **Create Acronym**  
   acronym(phrase) – Return the acronym in uppercase. ("Random Access Memory" → "RAM")
3. **Palindrome Sentence Check**  
   is\_sentence\_palindrome(s) – Ignore spaces/punctuation and check if it's a palindrome.
4. **Reverse Repeating Letters**  
   collapse\_repeats(word) – Replace sequences of repeated letters with one. ("boook" → "bok")
5. **Count Consonants**  
   count\_consonants(word) – Return the number of consonants in a word.
6. **Alternate Caps**  
   alternate\_caps(word) – Return word with alternating upper/lowercase letters. ("hello" → "HeLlO")
7. **Sort by Word Length**  
   sort\_by\_length(words) – Sort and return words from shortest to longest.
8. **Capitalize Titles**  
   capitalize\_title(title) – Capitalize words in a title except short words like "the", "of", "and".
9. **Make Histogram from Letters**  
   letter\_histogram(word) – Print letters as \* bar chart.

### 📦 ****List & Dictionary Manipulations****

1. **Rotate List Right**  
   rotate\_right(lst) – Move the last element to the front.
2. **Frequency Dictionary**  
   freq\_dict(lst) – Return dictionary with count of each item.
3. **Reverse Dictionary**  
   reverse\_dict(d) – Swap keys and values in a dictionary.
4. **Remove Every N-th Element**  
   remove\_every\_nth(lst, n) – Remove every n-th item from list.
5. **Mirror List**  
   mirror\_list(lst) – Return list mirrored around the center.
6. **Flatten 2D List**  
   flatten(matrix) – Return a flat list from a 2D list.
7. **Count Repeats**  
   count\_repeats(word) – Return dictionary of letters repeated more than once.
8. **Sort by Frequency**  
   sort\_by\_frequency(lst) – Return list sorted by frequency of each element.
9. **Remove Empty Strings**  
   clean\_list(lst) – Remove empty strings or whitespace-only strings.

### 🔐 ****Validation, Games, and Algorithms****

1. **Password Strength Checker**  
   check\_password\_strength(pw) – Return "Weak", "Medium", or "Strong" based on rules.
2. **Validate Email Format**  
   is\_valid\_email(email) – Return True if string is valid email format.
3. **Validate Date Format**  
   is\_valid\_date(date\_str) – Return True if string matches DD-MM-YYYY.
4. **Caesar Cipher Encoder**  
   caesar\_cipher(text, shift) – Shift every letter by shift positions.
5. **Is Prime (Efficient)**  
   is\_prime(n) – Return True if n is a prime number using efficient method.
6. **Roman Numeral to Integer**  
   roman\_to\_int(s) – Convert Roman numeral (like "XIV") to integer (14).
7. **Word Ladder Step**  
   is\_one\_letter\_diff(w1, w2) – Return True if words differ by only one letter.
8. **Calculate Grade**  
   get\_grade(score) – Return "A", "B", "C" etc. based on percentage score.
9. **Merge Two Sorted Lists**  
   merge\_sorted(lst1, lst2) – Merge two sorted lists into one sorted list.
10. **All Unique Characters**  
    all\_unique(word) – Return True if the word has no repeating characters.