🔁 Loops in Python – Problem Set with Solutions

➕ 1. Counting Positive Numbers

🩹 Problem:

Count how many numbers are **positive** in the given list.

```
numbers = [1, -2, 3, -4, 5, 6, -7, -8, 9, 10]
```

☑️ Solution:

```
count = 0
for num in numbers:
    if num > 0:
        count += 1

print("Positive numbers:", count)  # Output: 6
```

🔢 2. Sum of Even Numbers

🩹 Problem:

Find the sum of all even numbers from 1 to `n`.

☑️ Solution:

```
n = 10
total = 0
for i in range(1, n + 1):
    if i % 2 == 0:
        total += i

print("Sum of even numbers:", total)  # Output: 30
```

❌ 3. Multiplication Table Printer (Skip 5)

🩹 Problem:

Print table of 7 but **skip** iteration when multiplier is 5.

☑ Solution:

```
for i in range(1, 11):
    if i == 5:
        continue
    print(f"7 x {i} = {7 * i}")
```

🔁 4. **Reverse a String (No slicing!)**

🩹 Problem:

Reverse a string using a loop.

☑ Solution:

```
s = "Python"
rev = ""
for char in s:
    rev = char + rev  # prepend

print("Reversed:", rev)  # Output: nohtyP
```

🔍 5. **First Non-Repeated Character**

🩹 Problem:

Return first non-repeating character in a string.

☑ Solution:

```
s = "aabbcdeff"
for char in s:
    if s.count(char) == 1:
        print("First non-repeated:", char)  # Output: c
        break
```

🧮 6. **Factorial Calculator**

🩹 Problem:

Use `while` loop to calculate factorial.

☑ **Solution:**

```
n = 5
fact = 1
while n > 0:
    fact *= n
    n -= 1

print("Factorial:", fact)  # Output: 120
```

🔄 7. **Validate Input Between 1–10**

🩹 **Problem:**

Prompt until valid input is entered.

☑ **Solution:**

```
while True:
    num = int(input("Enter a number (1-10): "))
    if 1 <= num <= 10:
        print("Valid input:", num)
        break
    else:
        print("Try again!")
```

🔢 8. **Prime Number Checker**

🩹 **Problem:**

Check if a number is **prime**.

☑ **Solution:**

```
n = 17
is_prime = True

if n <= 1:
    is_prime = False
else:
    for i in range(2, int(n**0.5)+1):
        if n % i == 0:
            is_prime = False
            break
```

```
print("Is Prime:", is_prime)  # Output: True
```

📋 9. **List Uniqueness Checker**

🩹 Problem:

Detect if there are duplicates in a list.

```
items = ["apple", "banana", "orange", "apple", "mango"]
```

☑️ Solution:

```
seen = set()
for item in items:
    if item in seen:
        print("Duplicate found:", item)  # Output: apple
        break
    seen.add(item)
```

⏳ 10. **Exponential Backoff Retry**

🩹 Problem:

Print backoff times for 5 retries: `1, 2, 4, 8, 16`

☑️ Solution:

```
wait = 1
retries = 0

while retries < 5:
    print(f"Retry {retries+1}: wait {wait}s")
    wait *= 2
    retries += 1
```

📓 Part 1: 30+ Practice Problems – Loops Mastery

☑️ 11. Print All Odd Numbers Between 1 and 50

```
for i in range(1, 51):
    if i % 2 != 0:
        print(i, end=' ')
```

☑ 12. Count Digits in an Integer

```
num = 123456
count = 0
while num > 0:
    num //= 10
    count += 1
print("Digits:", count)  # Output: 6
```

☑ 13. Sum of Digits of a Number

```
n = 12345
sum_digits = 0
while n > 0:
    sum_digits += n % 10
    n //= 10
print("Sum of digits:", sum_digits)  # Output: 15
```

☑ 14. Palindrome Number Checker

```
num = 121
temp = num
rev = 0
while num > 0:
    rev = rev * 10 + num % 10
    num //= 10

print("Palindrome:", rev == temp)
```

☑ 15. Armstrong Number Checker (3-digit)

```
num = 153
temp = num
sum_cubes = 0
while num > 0:
```

```
    digit = num % 10
    sum_cubes += digit ** 3
    num //= 10

print("Armstrong:", sum_cubes == temp)
```

☑ 16. Sum of Squares from 1 to n

```
n = 5
total = 0
for i in range(1, n+1):
    total += i ** 2
print("Sum of squares:", total)
```

☑ 17. Find GCD of Two Numbers

```
a, b = 48, 60
while b != 0:
    a, b = b, a % b
print("GCD:", a)
```

☑ 18. Find LCM of Two Numbers

```
a, b = 15, 20
gcd = a
while b != 0:
    gcd, b = b, gcd % b
lcm = a * 20 // gcd
print("LCM:", lcm)
```

☑ 19. Print Fibonacci Series up to n terms

```
n = 10
a, b = 0, 1
for _ in range(n):
    print(a, end=' ')
    a, b = b, a + b
```

☑ 20. Count Vowels in a String

```
s = "Python is powerful"
vowels = "aeiouAEIOU"
count = 0
for ch in s:
    if ch in vowels:
        count += 1
print("Vowel count:", count)
```

👩‍💻 Mini Projects Using Loops

🔢 1. Number Guessing Game (1–100)

```
import random
target = random.randint(1, 100)
while True:
    guess = int(input("Guess (1-100): "))
    if guess == target:
        print("🎉 Correct!")
        break
    elif guess < target:
        print("Too low!")
    else:
        print("Too high!")
```

🔐 2. Password Strength Checker

```
password = input("Enter password: ")
has_upper = has_digit = has_special = False
for ch in password:
    if ch.isupper():
        has_upper = True
    elif ch.isdigit():
        has_digit = True
    elif not ch.isalnum():
        has_special = True

if has_upper and has_digit and has_special:
    print("Strong password 👍")
else:
    print("Weak password 🚨")
```

🧩 Pattern Practice (Nested Loops)

◇ 1. Right-Angled Triangle Pattern

```
n = 5
for i in range(1, n+1):
    print("*" * i)
```

```
*
**
***
****
*****
```

△ 2. Pyramid Pattern

```
n = 5
for i in range(1, n+1):
    print(" " * (n - i) + "*" * (2*i - 1))
```

```
    *
   ***
  *****
 *******
*********
```

■ 3. Number Square

```
n = 4
for i in range(1, n+1):
    for j in range(1, n+1):
        print(j, end=" ")
    print()
```

```
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
```

🧠 MCQs & Quiz (Loops Logic)

❓ Q1: What does `range(5)` return?

- A) [1, 2, 3, 4, 5]
- B) [0, 1, 2, 3, 4] ☑
- C) [0, 1, 2, 3, 4, 5]
- D) [1, 2, 3, 4]

❓ Q2: What is the output?

```
x = 0
while x < 3:
    x += 1
print(x)
```

- A) 0
- B) 3 ☑
- C) 2
- D) Infinite Loop

❓ Q3: Which loop guarantees at least one execution?

- A) `for`
- B) `while`
- C) `do-while`
- D) None ☑ (Python doesn't have `do-while`)

☑ Summary Cheatsheet

Concept	Loop Used	Example
Count / Sum	`for`	`sum += i`
Infinite input	`while`	`while True:`
Early Exit	`break`	`if x==target: break`
Skip Iteration	`continue`	`if i%2==0: continue`
Nesting	both	Loops inside loops (Patterns, Grids)