📘 Assignment: Loops and Iterations in Python

Part 1: Basics of `for` Loops

1. Print numbers from **1 to 10** using a `for` loop.
2. Print all **even numbers between 1 and 50**.
3. Print the **multiplication table** of a number (e.g., `5 × 1 = 5`, ...).
4. Write a program to **calculate the sum of first 100 natural numbers**.
5. Print each character of a string `"Python"` using a `for` loop.

Part 2: Basics of `while` Loops

1. Print numbers from **1 to 10** using a `while` loop.
2. Keep asking the user for a number until they enter **0** (then stop).
3. Write a program that prints numbers **10 down to 1** using a `while` loop.
4. Create a program that keeps asking for a password until the correct one (`"Python123"`) is entered.
5. Write a program that keeps rolling a dice (random 1–6) until it rolls a `6`.

Part 3: Loop Control (break, continue, pass)

1. Print numbers from 1 to 20, but **skip multiples of 5**.
2. Ask the user to enter numbers. Stop when they enter a **negative number**, then print the sum of all entered positive numbers.
3. Write a program that checks if a number is **prime** (use a loop).
4. Print the **first 10 Fibonacci numbers** using a loop.
5. Write a program that finds the **factorial of a number** using a loop.

Part 4: Nested Loops

1. Print a **square pattern** of stars () with size `5×5`.

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

1. Print a **right-angled triangle** of stars with 5 rows.

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

1. Print a **multiplication table (1 to 10)** in grid format.
2. Write a program that prints all pairs `(i, j)` where `i` and `j` are numbers from `1 to 3`.
3. Create a **number pyramid** like this (for `n=5`):

```
1
12
123
1234
12345
```

Part 5: Real-Life Scenarios

1. Simulate a **bank account system**:

- Start with `balance = 1000`.
- Keep asking deposit/withdraw until user types `"exit"`.
- Update balance after each operation.

1. Write a **guessing game**:

- Computer picks a random number (1–20).
- User keeps guessing until correct.
- Give hints `"Too High"` / `"Too Low"`.

1. Write a program that counts how many **vowels** are in a string.
2. Create a simple **ATM PIN system**:

- User gets max **3 attempts**.
- If PIN is correct → `"Access Granted"`.
- If wrong 3 times → `"Card Blocked"`.

1. Write a program that reads a list of numbers and finds the **largest and smallest number** using a loop (don't use `max` / `min`).

Part 6: Challenges (For Extra Practice 🚀)

1. Print a **diamond pattern** of stars (n=5).

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

1. Generate the **first 20 prime numbers** using a loop.

2. Build a **rock-paper-scissors game** using loops (play until user types `"quit"`).
3. Create a **basic calculator** that keeps running until user types `"exit"`.
4. Write a program to simulate a **snake game counter**:

- Start at score 0.
- Every loop, randomly decide if the snake eats food → +10 points.
- Stop the game when score reaches **100**.