LIST - TRAINING TASKS

- I will only accept tasks submitted **by the deadline**.
- I'll give you feedback on whether the solution is correct or not. The most important thing to me in your code is the **output**.
- The program **must display** some information (e.g., popups or messages) when it receives input or is running in a loop, indicating what action the user should take.
- It will **not** be possible to resend the code.
- An incorrect solution may negatively affect your activity grade.

1. Bank Account Management System with Transactions

📌 **Scenario:** A **bank** needs a system to manage **customer accounts**, allowing them to **deposit, withdraw, and check balances**.

📌 **Concepts Used:**
✅ **Dictionary for customer accounts**
✅ **Tuple for transaction history (immutable records)**
✅ **While loop for continuous transactions**
✅ **Functions for structured operations**

Problem Statement:

- Each customer has **an account balance** and **transaction history**.
- Customers can **deposit, withdraw, or check balance**.
- The **system records all transactions**.

Step 1: Define Customer Accounts

```
accounts = {
    "Alice": {"balance": 1000.0, "history": []},
    "Bob": {"balance": 1500.0, "history": []},
```

}

1. ATM Withdrawal System (Boolean, If-Else)

📌 **Scenario:** A **banking** system allows users to withdraw money based on certain conditions.

📌 **Concepts Used:**
✅ **Boolean**
✅ **If-Else conditions**
✅ **Function for withdrawal logic**

Problem Statement:

- A user can withdraw money only if their **account balance is sufficient** and the **withdrawal amount is a multiple of 10**.
- If either condition is not met, the withdrawal request should be denied.

Step 1: Define bank account data

```
accounts = {

   "Alice": {"balance": 1000},

   "Bob": {"balance": 250},

   "Charlie": {"balance": 80},

}
```

2. Inventory Management System (Boolean, If-Else, Dictionary)

📌 **Scenario:** Your company needs an inventory system to track the availability of products for sale. It will check if there is sufficient stock available before processing an order.

📌 **Concepts Used:**
✅ **Boolean logic**
✅ **If-Else conditions**
✅ **Dictionary to store inventory data**
✅ **For loop for inventory check**

Problem Statement:

- If the stock of an item is **greater than or equal to** the quantity requested, the order can be **fulfilled**.
- If not, the order will be **rejected** due to insufficient stock.

Step 1: Define inventory data

```
inventory = {
    "Laptop": 5,
    "Mouse": 10,
    "Keyboard": 8,
    "Headphones": 3,
}
```

3. Library Book Checkout System (Boolean, While Loop, Dictionary, If-Else)

📌 **Scenario:** A **library** allows users to check out books. If a book is already checked out, it cannot be checked out again. Track the **availability** of books and **allow or deny checkout** based on availability.

📌 **Rules:**
✅ If the book is **available**, it can be checked out.
✅ If the book is **already checked out**, deny the checkout request.

📌 **Concepts Used:other**
✅ **Boolean logic for availability check**
✅ **While loop for multiple user attempts**
✅ **If-Else conditions**
✅ **Dictionary to track book availability**

```
library_books = {
    "The Great Gatsby": True,
```

```
    "1984": False,

    "Moby Dick": True,

    "To Kill a Mockingbird": False

}
```

4. Student Grade Evaluation System (Boolean, If-Else, Dictionary, List)

📌 **Scenario:** Your school wants to evaluate student performance based on their **exam grades** and **attendance**. The student's grade is calculated based on both exam results and their attendance rate.

📌 **Rules:**
✅ A student needs **at least 50% attendance** to pass.
✅ A student needs to score **at least 60%** in the exam to pass.
✅ A student is considered **failed** if their attendance is below 50% or their exam score is below 60%, even if they meet one of the conditions.

📌 **Concepts Used:**
✅ **Boolean logic** to combine conditions
✅ **If-Else conditions** for evaluation
✅ **Lists and dictionaries** to track students' information

```
students = [

    {"name": "Alice", "exam_score": 85, "attendance": 90},

    {"name": "Bob", "exam_score": 58, "attendance": 75},

    {"name": "Charlie", "exam_score": 45, "attendance": 80},

    {"name": "Diana", "exam_score": 92, "attendance": 95},

    {"name": "Ethan", "exam_score": 60, "attendance": 40}

]
```