🗡️

CPP-A21: File Handling (Binary File)

📝 Problem 1: Hospital Patient Record System (Binary File Handling)

📌 **Problem Statement:**

Design a **Hospital Patient Record System** where patient details are stored in a **binary file** `"patients.dat"` .

Each patient has:

- **Patient ID** (Unique ID)
- **Name**
- **Age**
- **Disease**
- **Room Number**

🔷 **Tasks:**

1. **Add a new patient record** to the binary file.
2. **Display all patient records.**
3. **Search for a patient by Patient ID**.
4. **Update a patient's disease or room number**.
5. **Delete a patient's record** from the binary file.

🔻 **Example Input & Output:**

```
Input:
Add Patient: [P101, "John Doe", 45, "Fever", 202]
Search Patient ID: P101
Update Patient ID: P101 (Change Room Number to 205)
```

```
Delete Patient ID: P101

Output:
Patient Found: John Doe (Age: 45, Disease: Fever, Room: 202)
Patient Room Updated Successfully!
Patient Record Deleted Successfully!
```

📝 Problem 2: Movie Ticket Booking System (Binary File Handling)

📌 Problem Statement:

Develop a **Movie Ticket Booking System** where ticket bookings are stored in `"tickets.dat"` as **binary records**.

Each booking has:

- **Booking ID** (Unique ID)
- **Movie Name**
- **Customer Name**
- **Number of Seats**

🔷 Tasks:

1. **Book a new movie ticket** and save it in the binary file.
2. **Display all ticket bookings**.
3. **Search for a booking by Booking ID**.
4. **Update the number of seats** for a booking.
5. **Cancel (delete) a booking** from the file.

🔻 Example Input & Output:

```
Input:
Add Booking: [B001, "Avengers", "Alice", 3]
Search Booking ID: B001
Update Booking ID: B001 (Change Seats to 4)
Delete Booking ID: B001
```

```
Output:
Booking Found: Avengers - Alice (Seats: 3)
Seats Updated Successfully!
Booking Canceled Successfully!
```

📝 Problem 3: Vehicle Registration System (Binary File Handling)

📌 Problem Statement:

You need to create a **Vehicle Registration System** that maintains vehicle records in `"vehicles.dat"`.

Each vehicle has:

- **Registration Number** (Unique ID)
- **Owner Name**
- **Model**
- **Year of Manufacture**

🔷 Tasks:

1. **Register a new vehicle** and store it in the binary file.
2. **Display all registered vehicles**.
3. **Search for a vehicle by Registration Number**.
4. **Update the owner's name or model.**
5. **Delete a vehicle's registration** from the file.

🔻 Example Input & Output:

```
Input:
Register Vehicle: [MH12AB1234, "Rahul", "Toyota", 2020]
Search Registration Number: MH12AB1234
Update Registration Number: MH12AB1234 (Change Owner to Ramesh)
Delete Registration Number: MH12AB1234

Output:
Vehicle Found: Toyota (2020) - Owner: Rahul
```

```
Owner Name Updated Successfully!
Vehicle Record Deleted Successfully!
```

📝 Problem 4: Grocery Store Billing System (Binary File Handling)

📌 Problem Statement:

You are developing a **Grocery Store Billing System** where purchase details are stored in `"bills.dat"` in binary format.

Each purchase record has:

- **Bill Number** (Unique ID)
- **Customer Name**
- **Items Purchased**
- **Total Amount**

🔷 Tasks:

1. **Generate a new bill** and store it in the binary file.
2. **Display all bills**.
3. **Search for a bill by Bill Number**.
4. **Update the total amount of a bill**.
5. **Delete a bill record** from the file.

🔻 Example Input & Output:

```
Input:
Add Bill: [B1001, "Amit", "Rice, Sugar", 1200]
Search Bill Number: B1001
Update Bill Number: B1001 (Change Total Amount to 1300)
Delete Bill Number: B1001

Output:
Bill Found: Amit (Items: Rice, Sugar - Total: 1200)
Total Amount Updated Successfully!
Bill Record Deleted Successfully!
```

📝 Problem 5: Banking System (Binary File Handling)

📌 Problem Statement:

Create a **Banking System** where customer account details are stored in `"accounts.dat"` in **binary format**.

Each account has:

- **Account Number** (Unique ID)
- **Customer Name**
- **Balance**

🔷 Tasks:

1. **Open a new bank account** and store it in the binary file.
2. **Display all accounts**.
3. **Search for an account by Account Number**.
4. **Update balance after deposit or withdrawal**.
5. **Delete an account** from the file.

🔻 Example Input & Output:

```
Input:
Open Account: [A101, "Suresh", 5000]
Search Account Number: A101
Update Account Number: A101 (Deposit 2000)
Delete Account Number: A101

Output:
Account Found: Suresh (Balance: 5000)
Deposit Successful! New Balance: 7000
Account Closed Successfully!
```

Happy Coding!