Scenario: You are building a secure banking system where a user can make a withdrawal. Before a withdrawal can be processed, it must pass several checks:

1. **User Authentication** - The user must provide the correct password (hashed) for authentication.
2. **Multi-Factor Authentication (MFA)** - The user must enter a valid MFA code to verify their identity.
3. **Sufficient Balance** - The user must have enough funds to cover the withdrawal amount.
4. **Daily Transaction Limit** - The withdrawal amount must be within the daily transaction limit.

Steps to Solve the Challenge:

1. **Create a function called `verifyPassword`**
    - Purpose: This function will compare the user's input password with the stored hashed password.
    - **Input:** User's entered password, hashed password from the system.
    - **Output:** Returns `true` if the passwords match, otherwise returns `false`

```
function verifyPassword(inputPassword, storedHashedPassword):
    if bcrypt.compare(inputPassword, storedHashedPassword) == true:
        return true
    else:
        return false
```

Create a function called `verifyMFA`

- Purpose: This function will compare the MFA code entered by the user with the correct code stored in the system.
- **Input:** User's entered MFA code, correct MFA code.
- **Output:** Returns `true` if the codes match, otherwise returns `false`

```
function verifyMFA(inputMfaCode, correctMfaCode):
    if inputMfaCode == correctMfaCode:
        return true
    else:
        return false
```

Create a function called `checkBalance`

- Purpose: This function will verify whether the user has sufficient balance in their account to proceed with the withdrawal.
- **Input:** The withdrawal amount, the user's current balance.
- **Output:** Returns `true` if the balance is sufficient, otherwise returns `false`

```
function checkBalance(balance, withdrawalAmount):
    if balance >= withdrawalAmount:
        return true
    else:
        return false
```

Create a function called `checkDailyLimit`

- Purpose: This function will ensure that the withdrawal amount does not exceed the daily withdrawal limit.
- **Input:** The withdrawal amount, the daily transaction limit.
- **Output:** Returns `true` if the withdrawal is within the limit, otherwise returns `false`.

```
function checkDailyLimit(withdrawalAmount, dailyLimit):
    if withdrawalAmount <= dailyLimit:
        return true
    else:
        return false
```

Create a function called `processWithdrawal`

- Purpose: This is the main function that will process the withdrawal by checking all the conditions sequentially.
- **Input:** User's entered password, MFA code, withdrawal amount, actual MFA code, user's balance, daily limit, and stored password hash.
- **Output:** Returns success message if all conditions pass, otherwise returns an error message.

Steps within `processWithdrawal`:

- First, **verify the password** by calling the `verifyPassword` function.
- If the password is incorrect, return "Transaction Failed: Incorrect password."
- Next, **verify MFA** by calling the `verifyMFA` function.
- If MFA fails, return "Transaction Failed: MFA failed."
- Then, **check the balance** by calling the `checkBalance` function.
- If the balance is insufficient, return "Transaction Failed: Insufficient balance."
- Finally, **check the daily limit** by calling the `checkDailyLimit` function.
- If the withdrawal exceeds the daily limit, return "Transaction Failed: Amount exceeds daily limit."

- If all checks pass, **deduct the withdrawal amount from the balance** and return "Transaction Successful."

```
function processWithdrawal(user, inputPassword, inputMfaCode, withdrawalAmount):
    if verifyPassword(inputPassword, user.hashedPassword) == false:
        return "Transaction Failed: Incorrect password."

    if verifyMFA(inputMfaCode, user.correctMfaCode) == false:
        return "Transaction Failed: MFA failed."

    if checkBalance(user.balance, withdrawalAmount) == false:
        return "Transaction Failed: Insufficient balance."

    if checkDailyLimit(withdrawalAmount, user.dailyLimit) == false:
        return "Transaction Failed: Amount exceeds daily limit."

    user.balance -= withdrawalAmount
    return "Transaction Successful! New Balance: " + user.balance
```

Challenge Questions:

1. **Password Authentication:**
    - Why is it important to store passwords in a hashed format? What security advantage does this provide over plain text passwords?
2. **Multi-Factor Authentication (MFA):**
    - How does implementing MFA enhance the security of the transaction process? What types of attacks does it help prevent?
3. **Balance Verification:**
    - Why is it necessary to check the account balance before allowing a withdrawal? What risks are involved if this step is skipped?
4. **Daily Transaction Limit:**
    - What purpose does the daily transaction limit serve? How does it help in preventing fraudulent or excessive withdrawals?
5. **Improvement:**
    - If you were to add extra features, such as fraud detection (e.g., detecting abnormal withdrawal patterns), how would you go about doing this? What additional data would you track to detect fraud?