**Intermediate-level** Python regex problems:

### Problem 1: Extract Valid Email Addresses

#### **Description:**

Write a Python program to extract all **valid email addresses** from a given text.  
An email is considered valid if it follows this pattern: - Starts with **alphanumeric characters** ([a-zA-Z0-9]) - May contain periods (.), underscores (\_), or hyphens (-) before the @ - Must contain an **@ symbol** - Domain name contains **alphanumeric characters**, possibly hyphens (-) - Ends with a **valid top-level domain** (e.g., .com, .org, .net, .edu, etc.)

#### **Example Input:**

text = "Contact us at support@example.com, john.doe123@company.org, or invalid-email@com. Also, try jane\_doe@domain.co.uk."

#### **Expected Output:**

['support@example.com', 'john.doe123@company.org', 'jane\_doe@domain.co.uk']

### Problem 2: Extract Hashtags from a Tweet

#### **Description:**

Write a Python program to extract all **hashtags** (#hashtag) from a given text.  
- A hashtag: - Starts with # - Followed by **letters, numbers, or underscores** (\_) - Cannot contain spaces or special characters (@, !, $, etc.)

#### **Example Input:**

tweet = "Learning #Python is fun! #coding #100DaysOfCode #Regex\_Challenge"

#### **Expected Output:**

['#Python', '#coding', '#100DaysOfCode', '#Regex\_Challenge']

### Problem 3: Validate a Strong Password

#### **Description:**

Write a Python program to check if a password is **strong** based on these rules: 1. At least **8 characters long** 2. Contains at least **one uppercase letter** (A-Z) 3. Contains at least **one lowercase letter** (a-z) 4. Contains at least **one digit** (0-9) 5. Contains at least **one special character** (@, $, !, %, \*, ?, &)

#### **Example Inputs & Outputs:**

passwords = ["WeakPass", "Str0ng@Pass", "NoSpecial1", "short!1", "Secure#123"]

#### **Expected Output:**

WeakPass -> Invalid  
Str0ng@Pass -> Valid  
NoSpecial1 -> Invalid  
short!1 -> Invalid  
Secure#123 -> Valid