1. Medium – Extract all valid email addresses from a block of text

Scenario: You've got customer feedback logs where people sometimes leave emails in random spots of their messages. You need to extract them. Key Regex Features:

- Character classes
- Quantifiers (+, *, {m,n})
- Escaping special characters

---

2. Medium-Hard – Extract dates in multiple formats

Scenario: You need to match dates in DD/MM/YYYY, YYYY-MM-DD, or Month DD, YYYY format for a document processing tool. Key Regex Features:

- Alternation (|)
- Grouping ()
-

Word boundaries \b

3. Hard – Validate strong passwords

Scenario: You need to ensure passwords meet these rules:

- At least 8 characters
- At least one lowercase letter, one uppercase letter, one number, and one special character Key Regex Features:
- Lookaheads (?=...)
- Anchors ^ and $

---

4. Very Hard – Parse nested HTML tags (simplified)

Scenario: You're scraping HTML and want to extract text inside <div> tags, but only the innermost ones without nested <div>s. Key Regex Features:

- Lazy quantifiers *?
- Negative lookahead (?!...)
- Dot-all mode re.S

(Note: For real HTML, you should use *BeautifulSoup*, but this is a regex brain teaser.)

---

5. Expert – Match mathematical expressions respecting parentheses balance

Scenario: You want to match valid mathematical expressions like `2*(3+5)` or `(a+b)/(c-d)`, ensuring parentheses are balanced up to a certain nesting depth (regex can't do infinite nesting). Key Regex Features:

- Recursive patterns using Python's `regex` module (not `re`)
- Named groups & subroutine calls
