

Regular Expressions (RegEx) Programming Assignment

Introduction

This assignment will test your understanding of Regular Expressions (RegEx) in Python. Each question will have a set of test cases, and your goal is to write the correct RegEx pattern to pass all the provided test cases. Your solutions will be evaluated based on how many test cases pass.

Important: You must use regular expressions only. If any other methods (e.g., string manipulation, built-in functions, if-else etc) are used in your solution, the marks for the program will be 0 irrespective of how many test cases pass.

Submission Instructions

- **File Naming:** Name your Python file with your roll number, e.g., `2321cs12.py`.
- **Function Naming:** For each question, you should define a function with the exact signature as provided in the problem statement.
- Each function should return `True` if the input matches the regular expression pattern and `False` if it doesn't.
- **Submission:** Submit your Python file (`.py`) containing all the required functions.

Questions

Question 1: Matching a Valid Email Address

Write a function `is_valid_email(email: str) -> bool` that matches a valid email address. The email should follow the pattern:

- It must start with one or more alphanumeric characters.
- It should be followed by the “@” symbol.

- Then, there should be a domain name (letters, digits, hyphens, no two consecutive .).
- It must end with a top-level domain like .com, .org, .edu, etc.
- You must also consider domain-name standards

Function Signature:

```
def is_valid_email(email: str) -> bool:
    pass
```

Test Cases:

- "john.doe@sub.domain.org" → True
- "alice@domain.co" → True
- "@missingusername.com" → False
- "missing@domain" → False

Question 2: Valid Phone Number

Write a function `is_valid_phone_number(phone: str) -> bool` to check if a phone number is valid.

- The number can start with a + followed by a country code (1 to 4 digits) or it can be simply a number without any country code.
- Country code can be followed by a space or hyphen.
- The main phone number should consist of 10 digits and may include spaces or hyphens between groups.
- Phone number or country code should not start with 0

Function Signature:

```
def is_valid_phone_number(phone: str) -> bool:
    pass
```

Test Cases:

- "+1-800-555-1234" → True
- "+44 20 7123 1234" → True
- "800-555-1234" → True
- "12345-abc-6789" → False

Question 3: Matching IPv4 Address

Write a function `is_valid_ip(ip: str) -> bool` to validate if a string is a valid IPv4 address.

- An IPv4 address consists of four octets, each ranging from 0 to 255.
- Octets should be separated by a period (.).

Function Signature:

```
def is_valid_ip(ip: str) -> bool:  
    pass
```

Test Cases:

- "192.168.1.1" → True
- "255.255.255.255" → True
- "0.0.0.0" → True
- "192.168.1" → False
- "999.999.999.999" → False

Question 4: Valid URL

Write a function `is_valid_url(url: str) -> bool` to validate a URL.

- It should start with `http://` or `https://`.
- Followed by a domain name, which may contain letters, digits, hyphens, and dots.
- Optionally end with a path, query parameters, or fragment identifier.

Function Signature:

```
def is_valid_url(url: str) -> bool:  
    pass
```

Test Cases:

- "https://www.example.com" → True
- "http://example.com/path/to/page?query=1#fragment" → True
- "ftp://example.com" → False
- "http://example@com" → False
- "https://sub-domain.example.com" → True
- "www.example.com" → False

Submission

- Please submit your .py file to <https://forms.gle/6PPyXvdMfqBt23mp6>
- Submission Deadline: 11th February, 2026, 11:59PM
- The function names should exactly match the ones provided in the question
- Write the all four functions within the same file only
- Any libraries other than *re* will not be imported
- Using if-else, loops etc (anything other than regex) will result in 0 score.
- Make sure to return True/False from the functions, DO NOT print anything!
- The test-cases mentioned in the question is for your debugging, the actually scoring will be based on hidden test-cases.