**Practice Problems for PAT3 and Intro to engg:**

**Regex:**

1. Write a Python code that takes a string as input and returns True if the string matches a phone number format (e.g., (123) 456-7890 or 123-456-7890), and False otherwise using REGEX.
2. Write a function that takes a string and extracts all dates in the format DD/MM/YYYY. For example, given the input "The event is on 12/11/2023 and the deadline is 25/10/2024", it should return ['12/11/2023', '25/10/2024']
3. Write a Python code that takes a string (representing a tweet) and extracts all the hashtags (words starting with #). Return them as a list of strings.
4. Write a function that takes a string as input and returns True if it contains a valid URL. A valid URL should start with http:// or https://, followed by www., and contain a domain and an optional path (e.g., <https://www.example.com/path>).

**Strings:**

1. Write a python code to input a string and convert it into a list and remove all the special characters and output it as another string (eg., “re-verse!” becomes “reverse”).
2. Write a Python code that swaps the case of all characters in a string and then sorts the characters alphabetically (eg.,“bAcD “ becomes “BCad”).

**Regex and strings:**

1. Write a program that takes a single string as input and checks for valid email addresses. A valid email address follows the format username@domain.extension, where:

- username consists of alphanumeric characters, underscores (\_), or dots (.).

- domain consists of alphanumeric characters and may include dots (.).

- extension is between 2 to 4 alphabetic characters.

- The program should output all the valid email addresses found in the string. If no valid email addresses are found, output "No valid emails".

Input:

Please contact us at support@example.com or info@service.net.

Output:

support@example.com

info@service.net

Input:

Invalid emails: john@doe, jane.doe@website, user@domain.c0m.

Output:

No valid emails

Input:

Emails: test.email@domain.com, \_user@sub.domain.org, name@domain.

Output:

test.email@domain.com

[\_user@sub.domain.org](mailto:_user@sub.domain.org)

1. Write a program that takes a single string as input and checks for valid dates in the format "DD/MM/YYYY" , where:

- DD is a valid day (01 to 31).

- MM is a valid month (01 to 12).

- YYYY is a valid year.

The program should output all the valid dates found in the string. If no valid dates are found, output "No valid dates".

Input:

The event is on 29/02/2024 and the deadline is 32/13/2023 or 15/08/2023.

Output:

29/02/2024

15/08/2023

Input:

We met on 30/04/2021 and 31/11/2021.

Output:

30/04/2021

Input:

Dates: 99/99/9999, 32/01/2021.

Output:

No valid dates