Experiment Write-up: Extracting Data from Text Using Regular Expressions

1. Objective: To develop a Python program that reads a text file and extracts specific types of data, such as Indian names, email addresses, phone numbers, and dates, using regular expressions.

2. Requirements:

- Python installed on the system.
- A text file containing mixed data.
- Understanding of regular expressions (re module in Python).

3. Theory: Regular Expressions (regex) allow pattern matching in text. They are widely used in data extraction tasks to identify and extract specific patterns like phone numbers, email addresses, and dates.

- **Indian Name Extraction:** Matches common Indian names with first and last name formats.
- **Email Extraction:** Matches standard email formats with @ and domain extensions.
- **Phone Number Extraction:** Supports Indian phone numbers, including country code +91.
- **Date Extraction:** Identifies dates in DD/MM/YYYY format.

4. Algorithm:

1. Import the re module.
2. Open and read the text file.
3. Define regex patterns for Indian names, email, phone number, and date extraction.
4. Use re.findall() to extract matching data.
5. Display the extracted data.

5. Program Code:

```
import re


def extract_data(filename):
 with open(filename, 'r') as file:
 text = file.read()
```

names = re.findall(r'\b([A-Z][a-z]+\s[A-Z][a-z]+)\b', text) # Extracting Indian names

emails = re.findall(r'[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+', text) phones =

re.findall(r'\+91[-.\s]?\d{10}|\d{5}[-.\s]?\d{5}', text) # Indian phone numbers

dates = re.findall(r'\b(0[1-9]|[12][0-9]|3[01])/(0[1-9]|1[0-2])/(\d{4})\b', text) # DD/MM/YYYY format

print("Extracted Indian Names:", names)

print("Extracted Emails:", emails)

print("Extracted Phone Numbers:", phones)

print("Extracted Dates:", ["/".join(date) for date in dates])

Run the extraction function with a sample text file

extract_data("sample.txt")

```
import re

def extract_data(a):
    with open(r"C:\Users\nimis\OneDrive\Desktop\New folder\a.txt", 'r') as file:
        text = file.read()

    names = re.findall(r'\b([A-Z][a-z]+\s[A-Z][a-z]+)\b', text)  # Extracting Indian names
    emails = re.findall(r'[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+', text)
    phones = re.findall(r'\+91[-.\s]?\d{10}|\d{5}[-.\s]?\d{5}', text)  # Indian phone numbers
    dates = re.findall(r'\b(0[1-9]|[12][0-9]|3[01])/(0[1-9]|1[0-2])/(\d{4})\b', text)  # DD/MM/YYYY format

    print("Extracted Indian Names:", names)
    print("Extracted Emails:", emails)
    print("Extracted Phone Numbers:", phones)
    print("Extracted Dates:", ["/".join(date) for date in dates])

# Run the extraction function with a sample text file
extract_data("sample.txt")
```

6. Expected Output:

Extracted Indian Names: ['Amit Kumar', 'Priya Sharma']

Extracted Emails: ['example@gmail.com', 'user123@yahoo.com']

Extracted Phone Numbers: ['+91 9876543210', '98765 43210']

Extracted Dates: ['25/12/2023', '01/01/2024']

7. Conclusion: The program successfully reads a text file and extracts specific data using regular expressions. It demonstrates the effectiveness of regex in text processing and pattern matching.