Name: Adeola Odunewu

ID No: 14882

Batch No: DS2311

Regular Expression Practice Questions

Question 1- Write a Python program to replace all occurrences of a space, comma, or dot with a colon.

Solution:

```
import re
def replace(inputTx):
    output_text = re.sub(r'[ ,.]', ':', inputTx)
    return output_text
# ExampleSample_Text = 'Python Exercises, PHP exercises.'
# print outputresult = replace(Sample_Text)print("Input Text: ", Sample_Text)print("Output Text:", result)
```

Input Text: Python Exercises, PHP exercises.

Output Text: Python:Exercises::PHP:exercises:

Question 2- Create a dataframe using the dictionary below and remove everything (commas (,), !, XXXX, ;, etc.) from the columns except words.

Solution:

```
import pandas as pd
import re
# Input dictionary
Dictionary = {'SUMMARY': ['hello, world!', 'XXXXX test', '123four, five:; six...']}df = pd.DataFrame(Dictionary)
# Define function to filter the text
def clean_text(column):
    filtered_text = column.apply(lambda x: re.sub(r'[^a-zA-Z\s]', '', x))
    return filtered_text
# Apply clean_text function to the 'SUMMARY' column
df['SUMMARY'] = clean_text(df['SUMMARY'])
# Display the result
print(df)
```

Question 3 Create a function in python to find all words that are at least 4 characters long in a string. The use of the re.compile() method is mandatory.

Solution:

```
import re
def searchWord(inputItem):
    pattern = re.compile(r'\b\w{4,}\b')

    # Using findall method
    result = pattern.findall(inputItem)
    return result
inputItem = 'Create a function in python to find all words that are at least 4 characters'
wordCount = searchWord(inputItem)
print(wordCount)
```

['Create', 'function', 'python', 'find', 'words', 'that', 'least', 'characters']

Question 4- Create a function in python to find all three, four, and five character words in a string. The use of the re.compile() method is mandatory.

Solution:

```
import re
def wordSearch(inputString):
    pattern2 = re.compile(r'\b\w{3,5}\b')
    # Using find all method
    result = pattern2.findall(inputString)
    return result
inputString = 'Create a function in python to find all three, four, and five character words in a string. The use of the re.compile() method is mandatory'countWord = wordSearch(inputString)print(countWord)
```

['find', 'all', 'three', 'four', 'and', 'five', 'words', 'The', 'use', 'the']

Question 5- Create a function in Python to remove the parenthesis in a list of strings. The use of the re.compile() method is mandatory.

Solution:

```
import re
def removeBracket(sampleText):
    pattern = re.compile(r'\(|\)')
```

```
    result = [pattern.sub('',s) for s in sampleText]

    return result


# Sample string
 sampleText = ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data Science World)", "Data (Scientist)"]
 # Calling the function
 display = removeBracket(sampleText)
 print(display)
```

Question 6- Write a python program to remove the parenthesis area from the text stored in the text file using Regular Expression.

Solution:

```
 import re
 def rMoveb(tList):
    pattern = re.compile(r'\([^)]*\)')

    engine = [pattern.sub('', item) for item in tList]
    return engine#Sample testsText = ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data Science World)", "Data (Scientist)"]
 #Calling functionoutput = rMoveb(sText)print(output)
['example ', 'hr@fliprobo ', 'github ', 'Hello ', 'Data ']
```

Question 7- Write a regular expression in Python to split a string into uppercase letters.

Solution:

```
 import re
 def upperCase(sList):
    #Using find all to identify upperclass
    slitUp = re.findall(r'[A-Z]',sList)

    engine = ''.join(slitUp)
    return engine
 sTxt = 'ImportanceOfRegularExpressionsInPython'output1 = upperCase(sTxt)print(output1)
IOREIP
```

Question 8- Create a function in python to insert spaces between words starting with numbers.

Solution:

```
import re
def insertSpace(textString):
    # Use re.sub to insert a space after numbers
    result = re.sub(r'(?<=[a-zA-Z])(?=\d)', ' ', textString)
    return result
# Example usage:sText = "RegularExpression1IsAn2ImportantTopic3InPython"
# Calling the functionresult_with_spaces = insertSpace(sText)print(result_with_spaces)
```

RegularExpression 1IsAn 2ImportantTopic 3InPython

Question 9- Create a function in python to insert spaces between words starting with capital letters or with numbers.import re

Solution:

```
def spaceNum(textR):
    # Use re.sub to insert a space after numbers
    result = re.sub(r'(?<=[a-zA-Z])(?=\d)|(?<=\d)(?=[a-zA-Z])', ' ', textR)
    return result
# Example usage:text_input = 'RegularExpression1IsAn2ImportantTopic3InPython'
# Calling the functionoutput = spaceNum(text_input)print(output)
```

RegularExpression 1 IsAn 2 ImportantTopic 3 InPython

Question 10- Use the github link below to read the data and create a dataframe. After creating the dataframe extract the first 6 letters of each country and store in the dataframe under a new column called first_five_letters

Solution:

```
import pandas as pd
# GitHub link to the dataset
github_link =
"https://raw.githubusercontent.com/dsrscientist/DSData/master/happiness_score_dataset.
csv"
# Converting to a DataFrame
df = pd.read_csv(github_link)
# Getting the first 6 letters of each country & store in a
 df['first_five_letters'] = df['Country'].str[:6]
# Display the DataFrame with the new column
print(df[['Country', 'first_five_letters']].head())
```

Question 11- Write a Python program to match a string that contains only upper and lowercase letters, numbers, and underscores.

Solution:

```
import re
# Define the model to match the characters
def matCh(sText):
    pattern = re.compile(r'^[a-zA-Z0-9_]+$')
    return bool(pattern.match(sText))#Sample TextsText = 'Hello123_World'if matCh(sText):
    print(f'The string "{sText}" matches')else:
    print(f'The string "{sText}" does not match')
```

The string "Hello123_World" matches

Question 12- Write a Python program where a string will start with a specific number.

Solution:

```
import re
def startNum(testS, speciNum):
    pattern = re.compile(f'^{speciNum}')
    return bool(pattern.match(str(speciNum)))
# Example usage:
testS = "123example"
speciNum = 123
if startNum(testS, specific_number):
    print(f'The string "{testS}" starts with the number {speciNum}.')else:
    print(f'The string "{testS}" does not start with the number {speciNum}.')
```

The string "123example" starts with the number 123.

Question 13- Write a Python program to remove leading zeros from an IP address

Solution:

```
import re
def removeZeros(ipAdd):
    # Use re.sub to remove leading zeros from each segment of the IP address
    result = re.sub(r'\b0+(\d+)\b', r'\1', ipAdd)
    return result
# Example usage:
ipAdd = "192.168.001.001"
# Call the function and print the result
resultZeros = removeZeros(ipAdd)
```

```
print("IP address without leading zeros:", resultZeros)
```

Question 14- Write a regular expression in python to match a date string in the form of Month name followed by day number and year stored in a text file.

Solution:

```
import re
# Defining date match
def dateMatch(textNp):
    pattern =
re.compile(r'\b(?:January|February|March|April|May|June|July|August|September|Octo
ber|November|December)\s+\d{1,2}(st|nd|rd|th)?\s+\d{4}\b')
    return pattern.search(textNp)#Example textsampTxt =  'On August 15th 1947 that India
was declared independent from British colonialism, and the reins of control were handed
over to the leaders of the Country'
#Input testingoutPt = dateMatch(sampTxt)
print(f'Match Date',outPt)
```

Question 15- Write a Python program to search some literals strings in a string.

Solution:

```
import re
#Defining function
def searchLt(inputText, seaWords):
    for word in seaWords:
        if re.search(re.escape(word), inputText):
            print(f'The word "{word}" is found in the text.')
        else:
            print(f'The word "{word}" is not found in the text.')
# Example usage:sText = 'The quick brown fox jumps over the lazy dog.'seaWords = ['fox',
'dog', 'horse']
# Call the function
searchLt(sText, seaWords)
```

Question 16- Write a Python program to search a literals string in a string and also find the location within the original string where the pattern occurs

Solution:

```
import re
def searchLitlocation(inputText, searchWord):
    pattern = re.compile(re.escape(searchWord))
```

```
    match = pattern.search(inputText)

    if match:
        print(f'The word "{searchWord}" is found in the text.')
        print(f'Location: Start={match.start()}, End={match.end()}')
    else:
        print(f'The word "{searchWord}" is not found in the text.')
# Example usage:
smpText = 'The quick brown fox jumps over the lazy dog.'searchWord = 'fox'
# Call the function
searchLitlocation(smpText, searchWord)
```

Question 17- Write a Python program to find the substrings within a string.

Solution:

```
import re
#Defining function
def find_substrings(inputText, pattern):
    #re.finditer is used to find all occurrences of the specified pattern in the sample text
    matches = re.finditer(pattern, inputText)
    locations = [(match.start(), match.end()) for match in matches]
    return locations
# Example usage:
sampleText = 'Python exercises, PHP exercises, C# exercises'searchPattern = 'exercises'
# Call the function
substringLocations = find_substrings(sampleText, searchPattern)
# Print the result
if substringLocations:
    print(f'The pattern "{searchPattern}" is found in the text at the following locations:')
    for start, end in substringLocations:
        print(f'Location: Start={start}, End={end}')
else:
    print(f'The pattern "{searchPattern}" is not found in the text.')
```

Question 18- Write a Python program to find the occurrence and position of the substrings within a string.

Solution:

```
import re
#Defining function
def findOccurPos(mainString, substring):
    return [(match.start(), match.end()) for match in re.finditer(substring, mainString)]
# Example usage
inputString = "Hello, Hello, Hello, how are you doing today?"
subFind = "Hello"
occurPosi = findOccurPos(inputString, subFind)

# Display results
if occurPosi:
    print(f"Occurrences of '{subFind}' found at positions:")
    for start_pos, end_pos in occurPosi:
        print(f"Start position: {start_pos}, End position: {end_pos}")
else:
    print(f"No occurrences of '{subFind}' found in the input string.")
```

Question 19- Write a Python program to convert a date of yyyy-mm-dd format to dd-mm-yyyy format.

Solution:

```
import re
# Defining the Function
def convertDate(inputDate):
    # Define a regular expression pattern for matching yyyy-mm-dd format
    pattern = re.compile(r'(\d{4})-(\d{2})-(\d{2})')

    # Use re.sub to replace the matched pattern with the desired format
    outputDate = re.sub(pattern, r'\3-\2-\1', inputDate)

    return outputDate
# Example usage
inputDate = "2023-01-15"
convertedDate = convertDate(inputDate)
print(f"Original date: {inputDate}")
print(f"Converted date: {convertedDate}")
```

Question 20- Create a function in python to find all decimal numbers with a precision of 1 or 2 in a string. The use of the re.compile() method is mandatory.

Solution:

```
import re
#Defining function
def findDnum(text):
    # Define a regular expression pattern for matching decimal numbers with precision 1 or 2
    pattern = re.compile(r'\b\d+\.\d{1,2}\b')

    # Use re.findall to find all matches in the input text
    decimal_numbers = re.findall(pattern, text)

    return decimal_numbers
# Example usage
sampleText = "01.12 0132.123 2.31875 145.8 3.01 27.25 0.25"
result = findDnum(sampleText)
print("Input Text:", sampleText)
print("Decimal Numbers:", result)
```

Question 21- Write a Python program to separate and print the numbers and their position of a given string.

Solution:

```
import re
#Defining function
def separateNum(inputString):
    # Define a regular expression pattern for matching numbers
    pattern = re.compile(r'\d+')

    # Use re.finditer to find all occurrences of numbers along with their positions
    numbersPositions = [(match.group(), match.start(), match.end()) for match in
re.finditer(pattern, inputString)]

    # Print the results
    print("Numbers and their positions:")
    for number, start_pos, end_pos in numbersPositions:
        print(f"Number: {number}, Start Position: {start_pos}, End Position: {end_pos}")
# Example usageinputString = "Hello, 123, how are you doing today? 456.78 is a floating-
point number."
```

```
separateNum(inputString)
```

Question 22- Write a regular expression in python program to extract maximum/largest numeric value from a string.

Solution:

```
import re
# Defining function
def extractMvalue(inputText):
    # Define a regular expression pattern for matching numeric values
    pattern = re.compile(r'\b\d+\b')

    # Use re.findall to find all numeric values in the input text
    numericValues = [int(match) for match in re.findall(pattern, inputText)]

    # Find the maximum numeric value
    maxValue = max(numericValues, default=None)

    return maxValue
# Example usage
sampleText = 'My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'
result = extractMvalue(sampleText)
print("Sample Text:", sampleText)
print("Maximum Numeric Value:", result)
```

Question 23- Create a function in python to insert spaces between words starting with capital letters.

Solution:

```
import re
#Defining function
def insertSpaces(inputText):
    # Use re.sub to insert spaces before words starting with capital letters
    spacedText = re.sub(r'([a-z])([A-Z])', r'\1 \2', inputText)

    # Capitalize the first letter of the modified text
    spacedText = spacedText.capitalize()

    return spacedText
# Example usage
```

```
sampleText = "RegularExpressionIsAnImportantTopicInPython"result =
insertSpaces(sampleText)
print("Sample Text:", sampleText)
print("Output Text:", result)
```

Sample Text: RegularExpressionIsAnImportantTopicInPython

Output Text: Regular expression is an important topic in python

Question 24- Python regex to find sequences of one upper case letter followed by lower case letters

Solution:

```
import re#Defining functiondef findSequences(inputText):
    pattern = re.compile(r'[A-Z][a-z]+')
    sequences = re.findall(pattern, inputText)
    return sequences
# Example usage
sampleText = "AbCdEfGhIjKlMnOpQrStUvWxYz"
result = findSequences(sampleText)
print("Sample Text:", sampleText)
print("Found Sequences:", result)
```

Question 25- Write a Python program to remove continuous duplicate words from Sentence using Regular Expression.

Solution:

```
import re
#Defining model
def removeDuplicates(sentence):
    # Use re.sub with a backreference to remove continuous duplicate words
    modifiedSentence = re.sub(r'\b(\w+)\s+\1\b', r'\1', sentence)

    return modifiedSentence
# Example usage
sampleText = "Hello hello world world"
result = removeDuplicates(sampleText)
print("Sample Text:", sampleText)
print("Modified Text:", result)
```

Question 26- Write a python program using RegEx to accept string ending with alphanumeric character.

Solution:

```
import re
# Defining function
def endingWithalphaNumeric(inputString):
    # Define a regular expression pattern for matching a string ending with an alphanumeric character
    pattern = re.compile(r'.*\w$')

    # Use re.match to check if the input string matches the pattern
    match = re.match(pattern, inputString)

    return bool(match)
# Example usage
sampleString = "Hello123"result = endingWithalphaNumeric(sampleString)
print("Sample String:", sampleString)
print("Ends with Alphanumeric:", result)
```

Question 27-Write a python program using RegEx to extract the hashtags.

Solution:

```
import re
# Defining function
def extractHashtags(inputText):
    # Define a regular expression pattern for matching hashtags
    pattern = re.compile(r'#\w+')

    # Use re.findall to find all matches in the input text
    hashtags = re.findall(pattern, inputText)

    return hashtags
# Example usage
sampleText = """RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetization as the same has rendered USELESS <ed><U+00A0><U+00BD><ed><U+00B1><U+0089> "acquired funds" No wo"""
result = extractHashtags(sampleText)
print("Sample Text:", sampleText)
print("Extracted Hashtags:", result)
```

Question 28- Write a python program using RegEx to remove <U+..> like symbols

Solution:

```
import re
# Defining function
def removeSymbols(inputText):
    # Define a regular expression pattern for matching symbols like <U+..>
    pattern = re.compile(r'<U\+[A-Fa-f0-9]+>')

    # Use re.sub to replace the matched pattern with an empty string
    cleanedText = re.sub(pattern, '', inputText)

    return cleanedText
# Example usage
sampleText = "@Jags123456 Bharat band on
28??<ed><U+00A0><U+00BD><ed><U+00B8><U+0082>Those who  are protesting
#demonetization  are all different party leaders"result = removeSymbols(sampleText)
print("Sample Text:", sampleText)
print("Cleaned Text:", result)
```

Question 29- Write a python program to extract dates from the text stored in the text file.

Solution:

```
import re
def extractDate(file_path):
    # Read the content of the file
    with open(file_path, 'r') as file:
        content = file.read()

    # Define a regular expression pattern for matching dates in the format dd-mm-yyyy
    pattern = re.compile(r'\b\d{2}-\d{2}-\d{4}\b')

    # Use re.findall to find all matches in the file content
    dates = re.findall(pattern, content)

    return dates
# Example usage
```

```
file_path = r'C:\Users\Apelu\OneDrive\Desktop\File2 Regex Assessment DS2311\sample_text.txt'result = extractDate(file_path)
print("Dates extracted from the file:")
for date in result:
    print(date)
```

Question 30- Create a function in python to remove all words from a string of length between 2 and 4.

Solution:

```
import re
#Defining function
def removeWords(inputText):
    # Define a regular expression pattern for matching words of length between 2 and 4
    pattern = re.compile(r'\b\w{2,4}\b')

    # Use re.sub to replace the matched pattern with an empty string
    cleanedText = re.sub(pattern, '', inputText)

    return cleanedText
# Example usage
sampleText = "The following example creates an ArrayList with a capacity of 50 elements. 4 elements are then added to the ArrayList and the ArrayList is trimmed accordingly."result = removeWords(sampleText)
print("Sample Text:", sampleText)
print("Cleaned Text:", result)
```