Question 1- Write a Python program to replace all occurrences of a space, comma, or dot with a colon. Sample Text- 'Python Exercises, PHP exercises.' Expected Output: Python:Exercises::PHP:exercises:

```
import regex as re
string1='python Exercise, PHP exercise. '
replace=re.sub('\s|[,]|','":",string1)
print(replace)
```

```
python:Exercise::PHP:exercise.:
```

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

```
import pandas as pd
data={'SUMMARY' : ['hello, world!', 'XXXXX test', '123four, five:;
six...']}
df=pd.DataFrame(data)
df['SUMMARY']=df['SUMMARY'].str.replace('[^a-zA-Z\s]',"",regex=True)
print(df)
```

```
         SUMMARY
0    hello world
1     XXXXX test
2  four five six
```

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.

```
import regex as re
pattern=re.compile(r"\w{4}")
target_string="A Regular Expression is a sequence of character that
forms a search patter"
y=pattern.findall(target_string)
print(y)
```

```
['Regu', 'Expr', 'essi', 'sequ', 'ence', 'char', 'acte', 'that',
'form', 'sear', 'patt']
```

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.

```
import regex as re
pattern=re.compile(r"\w{3,5}")
target_string="Data science is the study of data to extract meaningful
insights for business. It is a multidisciplinary approach that
combines principles and practices from the fields of mathematics,
statistics, artificial intelligence, and computer engineering to
```

```
analyze large amounts of data. "
y=pattern.findall(target_string)
print(y)
```

```
['Data', 'scien', 'the', 'study', 'data', 'extra', 'meani', 'ngful',
'insig', 'hts', 'for', 'busin', 'ess', 'multi', 'disci', 'plina',
'appro', 'ach', 'that', 'combi', 'nes', 'princ', 'iples', 'and',
'pract', 'ices', 'from', 'the', 'field', 'mathe', 'matic', 'stati',
'stics', 'artif', 'icial', 'intel', 'ligen', 'and', 'compu', 'ter',
'engin', 'eerin', 'analy', 'large', 'amoun', 'data']
```

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.

```
import regex as re
pattern=re.compile(r"[()]|\s")
strings=["example (.com)", "hr@fliprobo (.com)", "github (.com)",
"Hello (Data Science World)", "Data (Scientist)"]
for string in strings:
    z=pattern.sub("",string)
    print(z)
# here we used various character,the character in the paranthesis
locates the value and the white space should be removed hence we use
the \s for that
```

```
example.com
hr@fliprobo.com
github.com
HelloDataScienceWorld
DataScientist
```

hence the expected results are matched with the results of mine.

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

```
import regex as re
with open('fliprobo.txt') as file: # stored file as fliprobo.txt
format
    for text in file:
        y=re.sub(r'\([^()]*\)',"",text) # here we \ to escape the
character in the sqaure brackets and checkes whether more occurance
are avaliable or not *
        print(y)

Sample_text=["example ", "hr@fliprobo ", "github ", "Hello ", "Data "]
```

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

```
import regex as re
string ="HelloStudentsOfDatascience"
pattern='[A-Z][^A-Z]*'
split=re.findall(pattern,string)
print(split)
```

```
['Hello', 'Students', 'Of', 'Datascience']
```

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

```
Sample_Text="RegularExpression1IsAn2ImportantTopic3InPython"
pattern='[0-9]'
ry=re.sub(pattern,r" ",Sample_Text)
print(ry)
```

```
RegularExpression IsAn ImportantTopic InPython
```

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

```
import regex as re
def insert_spaces(text):
    # for these we dont have the direct method for these so first lets
define a function
  pattern = '([A-Z][a-z0-9]+|\d+)'
  result = re.sub(pattern, r' \1', text)
  result = result.strip()# this strip function is use to remove extra
spaces
  return result
sample_text = "RegularExpression1IsAn2ImportantTopic3InPython"
output = insert_spaces(sample_text)
print(output)
```

```
Regular Expression1 Is An2 Important Topic3 In Python
```

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.

```
import pandas as pd
import regex as re
df=pd.read_csv('https://raw.githubusercontent.com/dsrscientist/DSData/
master/happiness_score_dataset.csv')
df
```

```
         Country                          Region  Happiness Rank  \
0    Switzerland                 Western Europe               1
1        Iceland                 Western Europe               2
2        Denmark                 Western Europe               3
```

```
3         Norway                     Western Europe                 4
4         Canada                      North America                 5
..           ...                                ...               ...
153       Rwanda                Sub-Saharan Africa                154
154        Benin                Sub-Saharan Africa                155
155        Syria  Middle East and Northern Africa                156
156      Burundi                Sub-Saharan Africa                157
157         Togo                Sub-Saharan Africa                158

     Happiness Score  Standard Error  Economy (GDP per Capita)
Family  \
0              7.587         0.03411                   1.39651
1.34951
1              7.561         0.04884                   1.30232
1.40223
2              7.527         0.03328                   1.32548
1.36058
3              7.522         0.03880                   1.45900
1.33095
4              7.427         0.03553                   1.32629
1.32261
..               ...             ...                       ...      ..
.
153            3.465         0.03464                   0.22208
0.77370
154            3.340         0.03656                   0.28665
0.35386
155            3.006         0.05015                   0.66320
0.47489
156            2.905         0.08658                   0.01530
0.41587
157            2.839         0.06727                   0.20868
0.13995

     Health (Life Expectancy)  Freedom  Trust (Government Corruption)
\
0                     0.94143  0.66557                        0.41978

1                     0.94784  0.62877                        0.14145

2                     0.87464  0.64938                        0.48357

3                     0.88521  0.66973                        0.36503

4                     0.90563  0.63297                        0.32957

..                        ...      ...                            ...

153                   0.42864  0.59201                        0.55191
```

```
154                    0.31910  0.48450                         0.08010

155                    0.72193  0.15684                         0.18906

156                    0.22396  0.11850                         0.10062

157                    0.28443  0.36453                         0.10731


     Generosity  Dystopia Residual
0       0.29678            2.51738
1       0.43630            2.70201
2       0.34139            2.49204
3       0.34699            2.46531
4       0.45811            2.45176
..          ...                ...
153     0.22628            0.67042
154     0.18260            1.63328
155     0.47179            0.32858
156     0.19727            1.83302
157     0.16681            1.56726

[158 rows x 12 columns]
```

```
# now we will pass the pattern that extract the first 6 letters of
each country
pattern=r'(^.\w{6})'
extract=df['Country'].str.extract(pattern, expand=True)
extract=extract.value_counts()
print(extract)
```

```
Afghani    1
Morocco    1
Paragua    1
Palesti    1
Pakista    1
          ..
Estonia    1
Ecuador    1
Dominic    1
Djibout    1
Zimbabw    1
Length: 86, dtype: int64
```

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

```
import regex as re
def match_string(string):
    pattern=r'^[a-zA-Z0-9_]+$'
```

```
    if re.match(pattern,string):
        print("Match Found")
    else:
        print("Match Not Found")
match_string("HelloAjit_123")
```

```
Match Found
```

```
# for another example
match_string("Hello Ajit_123")
```

```
Match Not Found
```

becasue we have made the pattern in such a way that it should contain only upper, lower case numbers and underscores but not white spaces

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

```
import regex as re
# it is as above question we should generate the pattern such that the string should start with numbers
def match_string(string):
    pattern=r'^[0-9]'
    if re.match(pattern,string):
        print("Match Found")
    else:
        print("Match Not Found")
match_string("123HelloAjit")
```

```
Match Found
```

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

```
import regex as re
ip = "192.158.01.308.078"
string = re.sub('\.[0]*', '.', ip)
print(string)
```

```
192.158.1.308.78
```

Here the only the leading 0's are matched so the 308 will remain as it is only 01 and 078

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

```
import regex as re
patterns=['fox','dog','horse']
sample_text='The quick brown fox jumps over the lazy dog.'
for pattern in patterns:
    print('searching',pattern,'in',sample_text)
    if re.search(pattern,sample_text):
```

```
        print("Match Found")
    else:
        print("The pattern not found")


searching fox in The quick brown fox jumps over the lazy dog.
Match Found
searching dog in The quick brown fox jumps over the lazy dog.
Match Found
searching horse in The quick brown fox jumps over the lazy dog.
The pattern not found
```

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

```
import regex as re
patterns='fox'
sample_text='The quick brown fox jumps over the lazy dog.'
z=re.search(patterns,sample_text)
print(z)

<regex.Match object; span=(16, 19), match='fox'>
```

Hence the fox word location is known from the span parameter starts with 16th loaction and ends with 18th.

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

```
import re
text = 'Python exercises, PHP exercises, C#'
pattern = 'exercises'
for match in re.findall(pattern, text):
    print('Found "%s"' % match)

Found "exercises"
Found "exercises"
```

hence the string contains only 2 exercises word so two substrings are found in a string.

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

```
import re
text = 'Python exercises, PHP exercises, C# exercises'
pattern = 'exercises'
for match in re.finditer(pattern, text):
    s = match.start()
    e = match.end()
    print('Found "%s" at %d:%d' % (text[s:e], s, e))
```

```
Found "exercises" at 7:16
Found "exercises" at 22:31
Found "exercises" at 36:45
```

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

```
date="2026-01-02"
new_date=re.sub(r'(\d{4})-(\d{1,2})-(\d{1,2})', '\\3-\\2-\\1', date)#
here we are taking groups of patterns
print('the updated date is :',new_date)

the updated date is : 02-01-2026
```

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.

```
import re
def find_decimal_numbers(string):
    pattern = re.compile(r'\d+\.\d{1,2}')
    decimal=re.findall(pattern,string)
    return decimal
sample_text = "01.12 0132.123 2.31875 145.8 3.01 27.25 0.25"
output = find_decimal_numbers(sample_text)
print(output)

['01.12', '0132.12', '2.31', '145.8', '3.01', '27.25', '0.25']
```

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

```
import re
text = "Hey my Name is Ajit I'm 22 years old"
for m in re.finditer("\d+", text):
    print(m.group(0))
    print("Index position:", m.start())

22
Index position: 24
```

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

```
import re
string='My marks in each semester are: 947, 896, 926, 524, 734, 950,
642'
number = re.findall('\d+', string)
```

```
number = map(int, number)# we use map to convert into integer
print("Max_value:",max(number))
```

```
Max_value: 950
```

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

```
import re
def insert_spaces(text):
    pattern = r'([A-Z][a-z]+)'
    space=re.sub(pattern,r' \1',text)
    space=space.strip()
    return space
sample_text = "RegularExpressionIsAnImportantTopicInPython"
output = insert_spaces(sample_text)
print(output)
```

```
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

```
import re
pattern=r'\b[A-Z][a-z]+\b'
text= "Hey this is Ajit I'm Studing Datascience"
letter=re.findall(pattern,text)
print(letter)
```

```
['Hey', 'Ajit', 'Studing', 'Datascience']
```

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

```
import re
pattern = r'\b(\w+)(\s+\1\b)+'
def remove_duplicates(text):
    duplicates=re.sub(pattern,r'\1',text)
    return duplicates
text="Hello hello world world"
output=remove_duplicates(text)
print(output)
```

```
Hello hello world
```

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

```
import re
pattern=r'\w[0-9]$'
def check_string(text):
    if(re.search(pattern,text)):
        print("the string contains Alphanumeric character")
    else:
        print("the string doesnot contain Alphanumeric character")
text="Ajit22"
output=check_string(text)
print(output)
```

```
the string contains Alphanumeric character
None
```

```
import re
pattern=r'\w[0-9]$'
def check_string(text):
    if(re.search(pattern,text)):
        print("the string contains Alphanumeric character")
    else:
        print("the string doesnot contain Alphanumeric character")
text="Ajit"
output=check_string(text)
print(output)
```

```
the string doesnot contain Alphanumeric character
None
```

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

```
import re

def extract_hashtags(text):
    hashtags = re.findall(r'#\w+', text)# (#\w+)matches the pattern
which contains#with word character
    return hashtags
text = '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'
hashtags = extract_hashtags(text)
print(hashtags)
```

```
['#Doltiwal', '#xyzabc', '#Demonetization']
```

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

```
import re
pattern=r'<U\+\w{4}>'
text="@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"
output=re.sub(pattern,"",text)
print(output)
```

```
@Jags123456 Bharat band on 28??<ed><ed>Those who  are protesting
#demonetization  are all different party leaders
```

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

```
import re
def remove_string(sample_text):
    pattern=re.compile(r'\b\w{2,4}\b')
    remove=re.sub(pattern,"",sample_text)
    return remove
sample_text="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."
output=remove_string(sample_text)
print(output)
```

```
 following example creates  ArrayList  a capacity   elements. 4
elements   added   ArrayList   ArrayList  trimmed accordingly.
```