In [ ]:
```
There are 30 solution of your assinment
NAME = MANOJ SINGODIYA
BATCH NO.= 2308
```

def replace_punctuation(text): punctuation = [' ', ',', '.'] for p in punctuation: text = text.replace(p, ':') return text

sample_text = 'Python Exercises, PHP exercises.' replaced_text = replace_punctuation(sample_text) print(replaced_text)

In [ ]:
```

```

In [2]:
```python
import re

def find_words(text):
    pattern = r'\b[aAeE]\w+\b'
    words = re.findall(pattern, text)
    return words

sample_text = 'Apple and elephant are animals, but eagle is a bird.'
words_starting_with_a_or_e = find_words(sample_text)
print(words_starting_with_a_or_e)
```

```
['Apple', 'and', 'elephant', 'are', 'animals', 'eagle']
```

In [3]:
```python
import re

def find_long_words(text):
    pattern = re.compile(r'\b\w{4,}\b')
    words = pattern.findall(text)
    return words

sample_text = 'The quick brown fox jumps over the lazy dog.'
long_words = find_long_words(sample_text)
print(long_words)
```

```
['quick', 'brown', 'jumps', 'over', 'lazy']
```

In [4]:
```python
import re

def find_words(text):
    # Compile the regular expression pattern
    pattern = re.compile(r'\b\w{3,5}\b')

    # Find all matches in the text
    matches = pattern.findall(text)

    return matches

# Sample text
text = "The quick brown fox jumps over the lazy dog."

# Call the function and print the result
output = find_words(text)
print(output)
```

```
['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog']
```

In [5]:
```python
import re
text = ["example (.com)", "hr@fliprobo (.com)","github (.com)", "Hello (Data Science world)","Data (Scientist)"]
list = re.compile(r"[()]")
for text in text:
    print(re.sub(list,"",text))
```

```
example .com
hr@fliprobo .com
github .com
Hello Data Science world
Data Scientist
```

In [6]:
```python
import re

def remove_parentheses(strings):
    pattern = re.compile(r'\((.*?)\)')
    updated_strings = [pattern.sub('', string) for string in strings]
    return updated_strings

sample_text = ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data Science World)", "Data (Scientist)"]
updated_text = remove_parentheses(sample_text)
print(updated_text)
```

```
['example ', 'hr@fliprobo ', 'github ', 'Hello ', 'Data ']
```

In [7]:
```python
import re

# Sample text
sample_text = "ImportanceOfRegularExpressionsInPython"

# Split the string into uppercase letters
result = re.findall(r'[A-Z][a-z]*', sample_text)

# Print the result
print(result)
```

```
['Importance', 'Of', 'Regular', 'Expressions', 'In', 'Python']
```

In [8]:
```python
import re

def insert_spaces(text):
    # Insert spaces between words starting with numbers
    updated_text = re.sub(r'(\d+)([A-Za-z]+)', r'\1 \2', text)
    return updated_text

# Sample text
sample_text = "RegularExpression1IsAn2ImportantTopic3InPython"

# Call the function and print the result
result = insert_spaces(sample_text)
print(result)
```

```
RegularExpression1 IsAn2 ImportantTopic3 InPython
```

In [9]:
```python
import re

def insert_spaces(text):
    # Insert spaces between words starting with capital letters or numbers
    updated_text = re.sub(r'([A-Z\d])([A-Za-z]+)', r'\1 \2', text)
    return updated_text

# Sample text
sample_text = "RegularExpression1IsAn2ImportantTopic3InPython"

# Call the function and print the result
result = insert_spaces(sample_text)
print(result)
```

```
R egularExpression1 IsAn2 ImportantTopic3 InPython
```

In [10]:
```python
sample_text = 'Sample Text- Hello my name is Data Science and my email address is xyz@domain.com and alternate email address is xyz.abc@sdomain.domain.com. Please contact us at hr@
pattern = (r"[a-z0-9\.\-+]+@[a-z0-9\.\-+]+\.[a-z]+")

matches = re.findall(pattern,sample_text)
print(matches)
```

```
['xyz@domain.com', 'xyz.abc@sdomain.domain.com', 'hr@fliprobo.com']
```

In [11]:
```python
import re

def is_valid_string(string):
    pattern = r'^[a-zA-Z0-9_]+$'
    return bool(re.match(pattern,string))

string = "Hello_World123"
result = is_valid_string(string)
print(result)
```

```
True
```

In [12]:
```python
def starts_with_number(string,number):
    return string.startswith(str(number))


string ="123abc"
number =123
result =starts_with_number(string,number)
print(result)
```

```
True
```

In [13]:
```python
def remove_leading_zeros(ip_address):
    return'.'.join(str(int(x))for x in
  ip_address.split('.'))

ip_address ="192.168.001.001"
result = remove_leading_zeros(ip_address)
print(result) # output:192.168.1.1
```

```
192.168.1.1
```

In [14]:
```
mhnm hm nh
```

```
  Cell In[14], line 1
    mhnm hm nh
         ^
SyntaxError: invalid syntax
```

In [15]:
```python
def search_words(text, words):
    found_words = []
    for word in words:
        if word in text:
            found_words.append(word)
    return found_words

text = 'The quick brown fox jumps over the lazy dog.'
searched_words = ['fox', 'dog', 'horse']
result = search_words(text, searched_words)
print(result)
```

```
['fox', 'dog']
```

In [16]:
```python
def search_words(text, words):
    found_words = []
    for word in words:
        if word in text:
            found_words.append(word)
    return found_words

text = 'The quick brown fox jumps over the lazy dog.'
searched_words = ['fox']
result = search_words(text, searched_words)
print(result)
```

```
['fox']
```

In [17]:
```python
import re

def find_substrings(text, pattern):
    substrings = re.findall(pattern, text)
    return substrings

text = 'Python exercises, PHP exercises, C# exercises'
pattern = 'exercises'
result = find_substrings(text, pattern)
print(result)
```

```
['exercises', 'exercises', 'exercises']
```

In [18]:
```python
def find_substrings(text, pattern):
    occurrences = []
    start = 0
    while True:
        index = text.find(pattern, start)
        if index == -1:
            break
        occurrences.append((pattern, index))
        start = index + 1
    return occurrences

text = 'Python exercises, PHP exercises, C# exercises'
pattern = 'exercises'
result = find_substrings(text, pattern)
print(result)
```

```
[('exercises', 7), ('exercises', 22), ('exercises', 36)]
```

In [19]:
```python
from datetime import datetime

def convert_date(date):
    # Convert the input string to a datetime object
    datetime_obj = datetime.strptime(date, '%Y-%m-%d')
    # Format the datetime object to the desired format
    formatted_date = datetime_obj.strftime('%d-%m-%Y')
    return formatted_date

date = '2023-08-28'
converted_date = convert_date(date)
print(converted_date)
```

```
28-08-2023
```

In [20]:
```python
import re

def find_decimal_numbers(text):
    pattern = re.compile(r'\b\d+\.\d{1,2}\b')
    decimal_numbers = pattern.findall(text)
    return decimal_numbers

text = "01.12 0132.123 2.31875 145.8 3.01 27.25 0.25"
decimal_numbers = find_decimal_numbers(text)
print(decimal_numbers)
```

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

In [21]:
```python
def separate_numbers(text):
    numbers = []
    for index, char in enumerate(text):
        if char.isdigit():
            numbers.append((char, index))
    return numbers

text = "Hello123World456"
number_positions = separate_numbers(text)
for number, position in number_positions:
    print(f"Number: {number}, Position: {position}")
```

```
Number: 1, Position: 5
Number: 2, Position: 6
Number: 3, Position: 7
Number: 4, Position: 13
Number: 5, Position: 14
Number: 6, Position: 15
```

In [22]:
```python
import re

def extract_maximum_numeric_value(text):
    numbers = re.findall(r'\d+', text)
    if numbers:
        maximum = max(map(int, numbers))
        return maximum
    else:
        return None

text = 'My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'
maximum_value = extract_maximum_numeric_value(text)
print(maximum_value)
```

```
950
```

In [23]:
```python
import re

def insert_spaces(text):
    spaced_text = re.sub(r'(?<=.)([A-Z][a-z]+)', r' \1', text)
    return spaced_text

text = "RegularExpressionIsAnImportantTopicInPython"
spaced_text = insert_spaces(text)
print(spaced_text)
```

```
Regular Expression Is An Important Topic In Python
```

In [24]:
```python
import re

text = "RegularExpressionIsAnImportantTopicInPython"
matches = re.findall(r'[A-Z][a-z]+', text)
print(matches)
```

```
['Regular', 'Expression', 'Is', 'An', 'Important', 'Topic', 'In', 'Python']
```

In [26]:
```python
import re

def remove_duplicates(sentence):
    pattern = r'\b(\w+)(\s+\1)+\b'
    cleaned_sentence = re.sub(pattern, r'\1', sentence)
    return cleaned_sentence

text = "Hello hello world world"
cleaned_text = remove_duplicates(text)
print(cleaned_text)
```

```
Hello hello world
```

In [28]:
```python
import re

# Sample text
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'

# Extract hashtags using regular expressions
hashtags = re.findall(r'#\w+', text)

# Print the extracted hashtags
print(hashtags)
```

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

In [29]:
```python
import re

# Sample text
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"

# Remove <U+..> symbols using regular expressions
clean_text = re.sub(r'<U\+\w+>', '', text)

# Print the cleaned text
print(clean_text)
```

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

In [30]:
```python
import re

def remove_words(text):
    pattern = re.compile(r'\b\w{2,4}\b')
    return pattern.sub('', text)

# Sample text
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."

# Remove words of length between 2 and 4
output = remove_words(text)

# Print the modified text
print(output)
```

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

In [ ]:
```

```