

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

Ans- import re

```
text = 'Python Exercises, PHP exercises.'
```

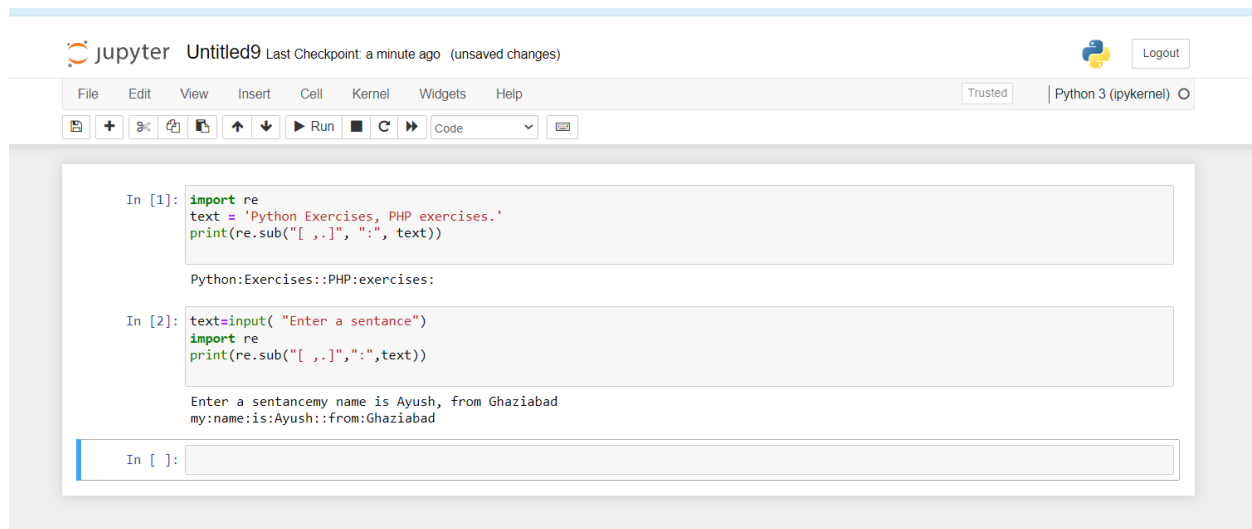
```
print(re.sub("[ ,.]", ":", text))
```

Another Example of if string take input from user. Then program will be run this form

```
text=input( "Enter a sentence")
```

```
import re
```

```
print(re.sub("[ ,.]", ":", text))
```



The screenshot shows a Jupyter Notebook titled 'Untitled9'. The first code cell (In [1]) contains the following code:

```
import re
text = 'Python Exercises, PHP exercises.'
print(re.sub("[ ,.]", ":", text))
```

The output of this cell is: Python:Exercises::PHP:exercises:

The second code cell (In [2]) contains the following code:

```
text=input( "Enter a sentence")
import re
print(re.sub("[ ,.]", ":", text))
```

The output of this cell is: Enter a sentancemy name is Ayush, from Ghaziabad
my:name:is:Ayush::from:Ghaziabad

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

Ans- def my_assignment1():

```
    print("Ends")
```

```
#another
```

```
text=input( "Enter a sentence")
```

```
import re
```

```
#text = 'Python Exercises, PHP exercises.'
```

```
print(re.sub("[ ,.]", ":", text))
```

```
#call the function
```

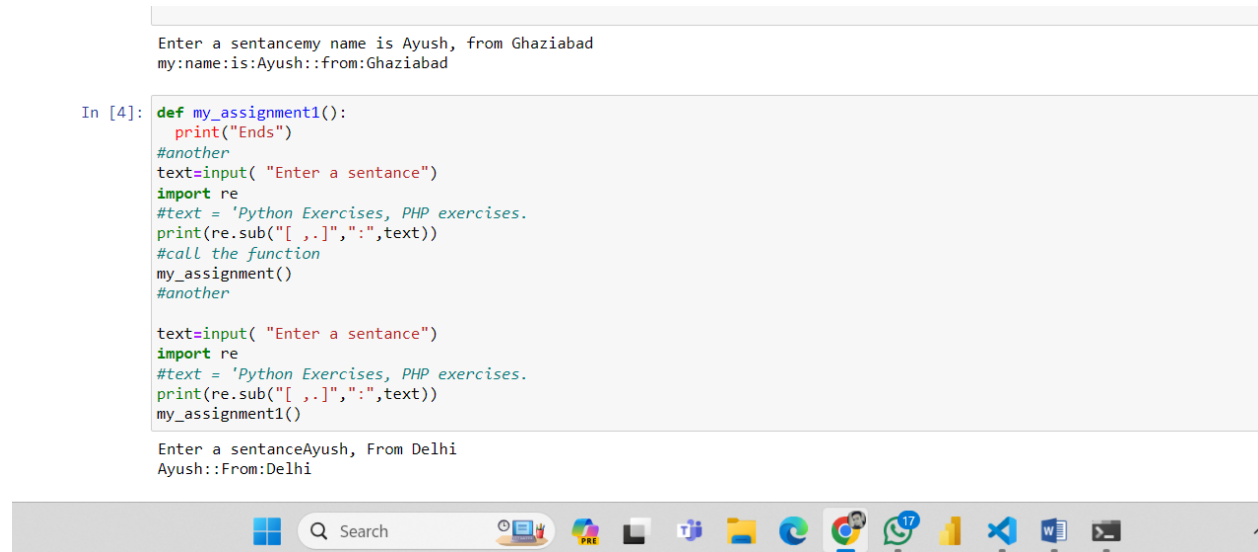
```
my_assignment()
```

```
#another
```

```

text=input( "Enter a sentence")
import re
#text = 'Python Exercises, PHP exercises.
print(re.sub("[ ,.]",":",text))
my_assignment1()

```



```

Enter a sentencemy name is Ayush, from Ghaziabad
my:name:is:Ayush::from:Ghaziabad

In [4]: def my_assignment1():
        print("Ends")
        #another
        text=input( "Enter a sentence")
        import re
        #text = 'Python Exercises, PHP exercises.
        print(re.sub("[ ,.]",":",text))
        #call the function
        my_assignment()
        #another

        text=input( "Enter a sentence")
        import re
        #text = 'Python Exercises, PHP exercises.
        print(re.sub("[ ,.]",":",text))
        my_assignment1()

Enter a sentenceAyush, From Delhi
Ayush::From:Delhi

```

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.

Ans-

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

Ans- # Define a function to return a string with unique words

```

def unique_list(text_str):
    # Split the input string into a list of words
    l = text_str.split()

    # Initialize an empty list to store unique words
    temp = []

    # Iterate through each word in the list
    for x in l:
        # Check if the word is not already in the temporary list
        if x not in temp:

```

```

        # If true, add the word to the temporary list
        temp.append(x)

    # Join the unique words into a string and return the result
    return ' '.join(temp)

# Initialize a string
text_str = "Hello hello world world"

# Print the original string
print("Original String:")
print(text_str)

# Print a newline for better formatting
print("\nAfter removing duplicate words from the said string:")

# Call the function to remove duplicate words and print the result
print(unique_list(text_str))

```

The screenshot shows a Python IDE with a file named `ayush1996.py` open. The code defines a function `unique_list` that takes a string `text_str` and returns a string with unique words. The function splits the input string into a list of words, iterates through each word, and checks if it is already in a temporary list. If not, it appends the word to the temporary list. Finally, it joins the unique words back into a string and returns it. The code also initializes a string `text_str = "Hello hello world world"`, prints the original string, and calls the `unique_list` function to remove duplicate words and print the result.

```

1 # Define a function to return a string with unique words
2 def unique_list(text_str):
3     # Split the input string into a list of words
4     l = text_str.split()
5
6     # Initialize an empty list to store unique words
7     temp = []
8
9     # Iterate through each word in the list
10    for x in l:
11        # Check if the word is not already in the temporary list
12        if x not in temp:
13            # If true, add the word to the temporary list
14            temp.append(x)
15
16    # Join the unique words into a string and return the result
17    return ' '.join(temp)
18
19 # Initialize a string
20 text_str = "Hello hello world world"
21
22 # Print the original string
23 print("Original String:")
24 print(text_str)
25
26 # Print a newline for better formatting
27 print("\nAfter removing duplicate words from the said string:")
28
29 # Call the function to remove duplicate words and print the result
30 print(unique_list(text_str))
31

```

The terminal output shows the original string and the result after removing duplicate words:

```

Original String:
Hello hello world world

After removing duplicate words from the said string:
Hello hello world

```

The status bar at the bottom indicates the current line and column (Ln 16, Col 64), the file encoding (UTF-8), the line ending (CRLF), the interpreter (Python 3.12.1 64-bit), and the system clock (7:13 PM, 4/28/2024).

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

```
def extract_hashtags(text):
    hashtags = re.findall(r'#\w+', text)
    return hashtags

# 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
hashtags = extract_hashtags(text)

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

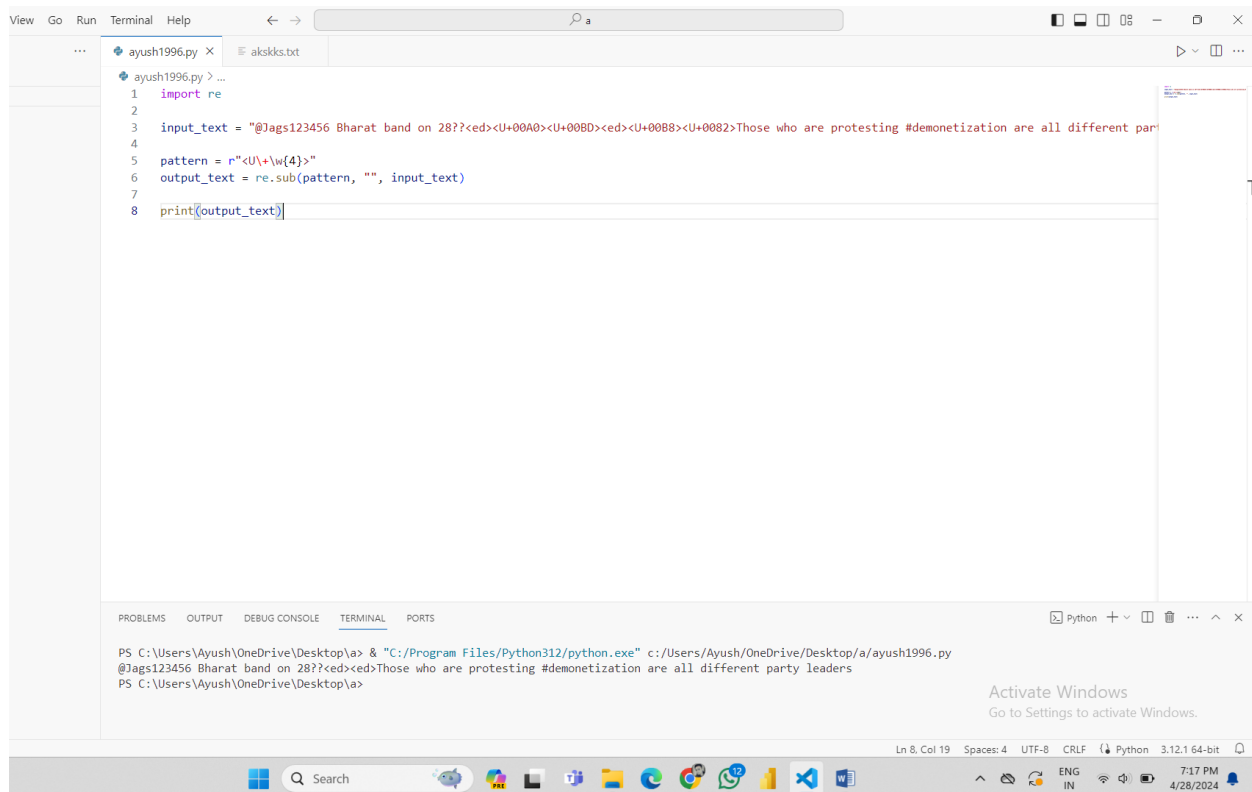
Question 28- Write a python program using RegEx to remove <U+..> like symbols

Check the below sample text, there are strange symbols something of the sort <U+..> all over the place. You need to come up with a general Regex expression that will cover all such symbols.

Ans- `import re`

```
input_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"
```

```
pattern = r"<U\+\w{4}>"  
output_text = re.sub(pattern, "", input_text)  
  
print(output_text)
```



The screenshot shows a Python IDE with a file named 'ayush1996.py' open. The code in the editor is as follows:

```
1 import re  
2  
3 input_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"  
4  
5 pattern = r"<U\+\w{4}>"  
6 output_text = re.sub(pattern, "", input_text)  
7  
8 print(output_text)
```

The terminal output at the bottom shows the command being run and the resulting string after the removal of the Unicode escape sequences:

```
PS C:\Users\Ayush\OneDrive\Desktop> & "C:/Program Files/Python312/python.exe" c:/Users/Ayush/OneDrive/Desktop/a/ayush1996.py  
@Jags123456 Bharat band on 28??<ed><ed>Those who are protesting #demonetization are all different party leaders  
PS C:\Users\Ayush\OneDrive\Desktop>
```

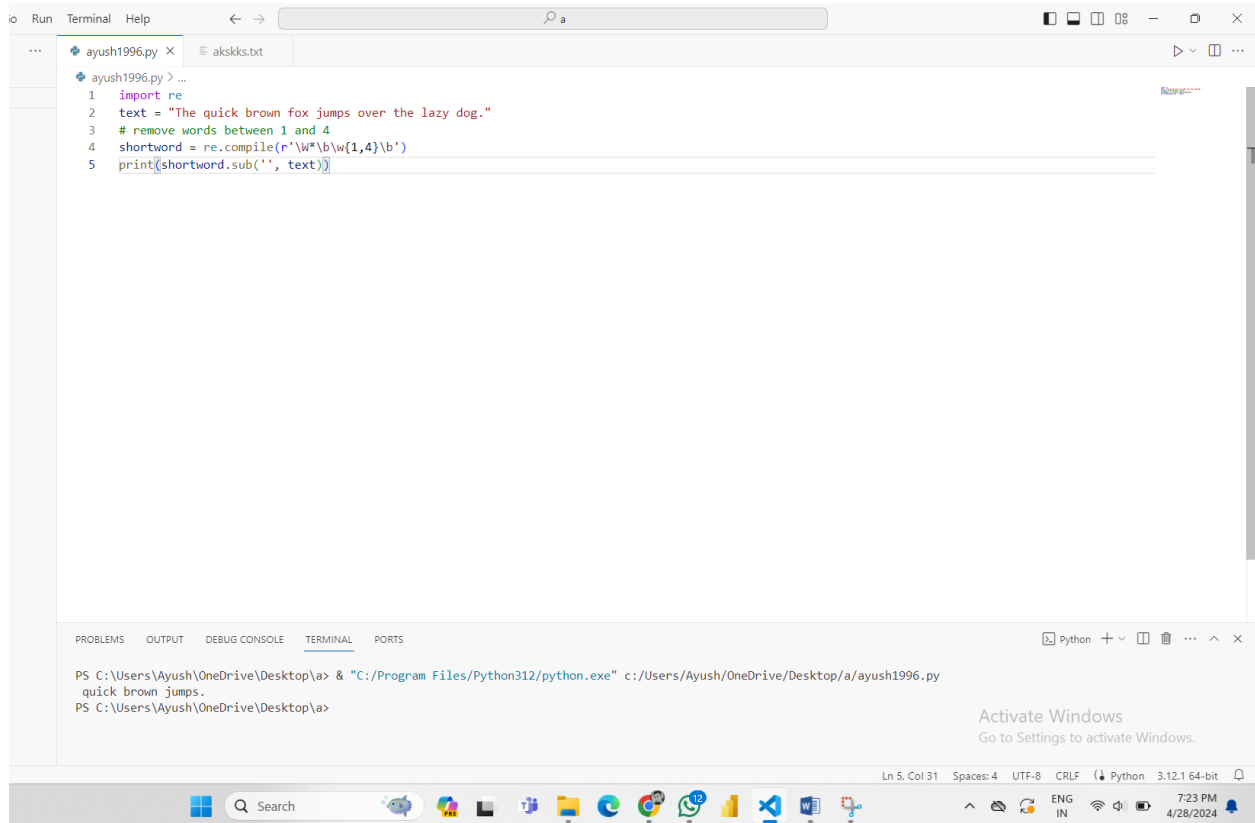
Question 30- Create a function in python to remove all words from a string of length between 2 and 4.

The use of the `re.compile()` method is mandatory.

Ans- `import re`

```
text = "The quick brown fox jumps over the lazy dog."  
# remove words between 1 and 4
```

```
shortword = re.compile(r'\W*\b\w{1,4}\b')
print(shortword.sub('', text))
```



The screenshot shows a Python IDE with a file named 'ayush1996.py'. The code in the editor is as follows:

```
1 import re
2 text = "The quick brown fox jumps over the lazy dog."
3 # remove words between 1 and 4
4 shortword = re.compile(r'\W*\b\w{1,4}\b')
5 print(shortword.sub('', text))
```

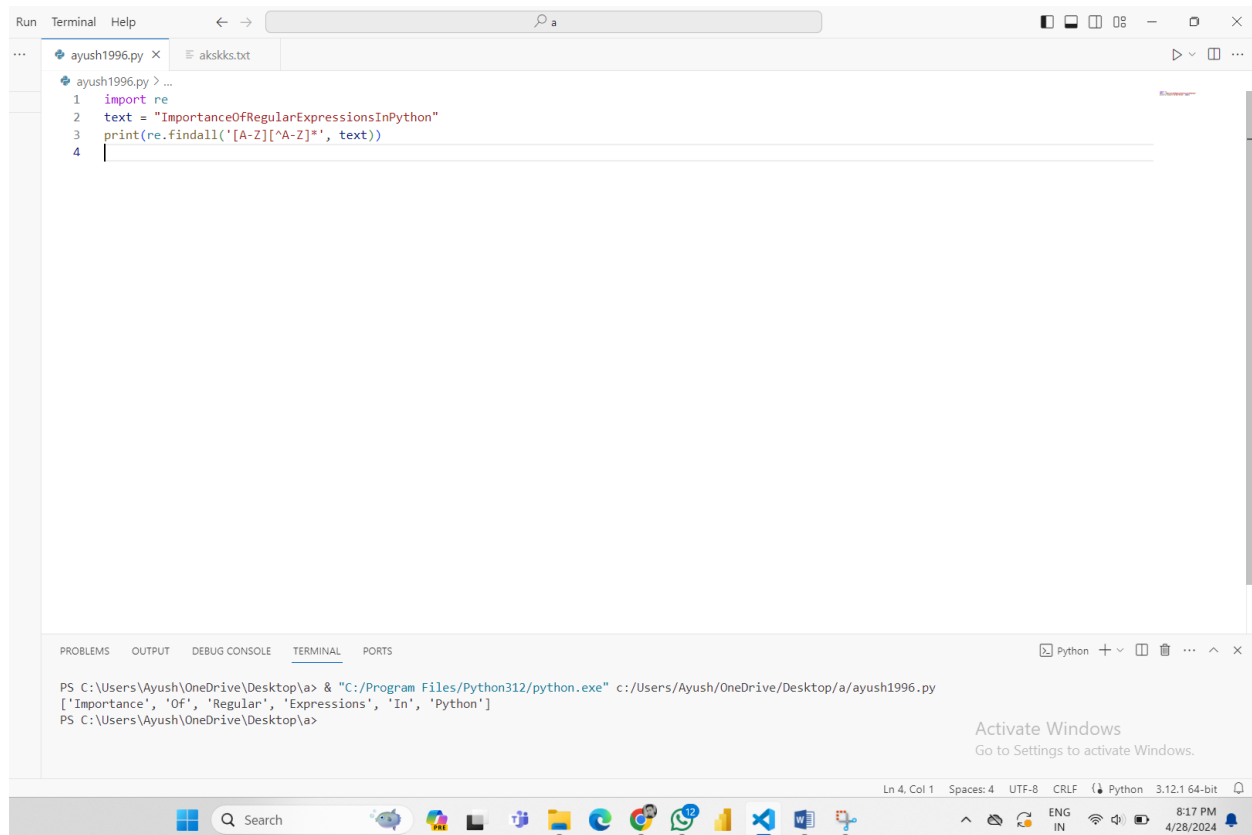
The terminal output at the bottom shows the command prompt running the script, resulting in the output: 'quick brown jumps.'.

```
PS C:\Users\Ayush\OneDrive\Desktop\> & "C:/Program Files/Python312/python.exe" c:/Users/Ayush/OneDrive/Desktop/a/ayush1996.py
quick brown jumps.
PS C:\Users\Ayush\OneDrive\Desktop\>
```

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

Ans- `import re`

```
text = "ImportanceOfRegularExpressionsInPython"
print(re.findall('[A-Z][^A-Z]*', text))
```



The screenshot shows a Visual Studio Code editor window with a Python file named `ayush1996.py` open. The code in the file is as follows:

```
1 import re
2 text = "ImportanceOfRegularExpressionsInPython"
3 print(re.findall('[A-Z][^A-Z]*', text))
4
```

The terminal at the bottom shows the command to run the script and its output:

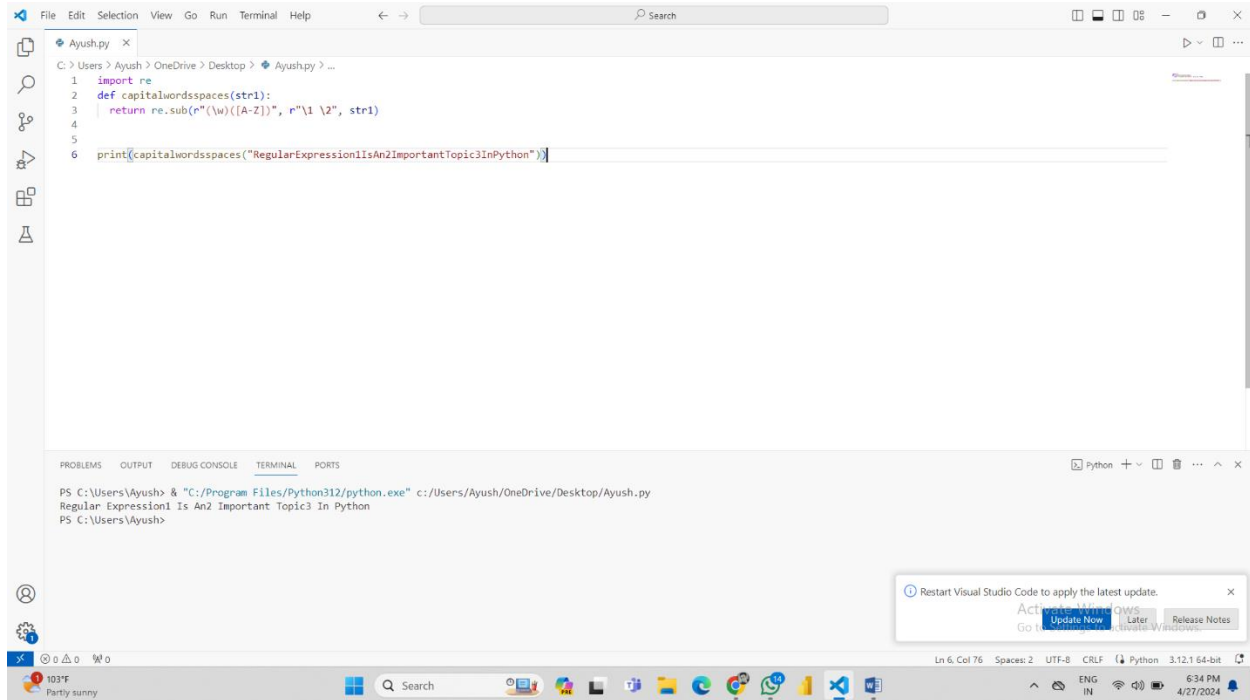
```
PS C:\Users\Ayush\OneDrive\Desktop\> & "C:/Program Files/Python312/python.exe" c:/Users/Ayush/OneDrive/Desktop/a/ayush1996.py
['Importance', 'Of', 'Regular', 'Expressions', 'In', 'Python']
PS C:\Users\Ayush\OneDrive\Desktop\>
```

The status bar at the bottom indicates the file is at line 4, column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and is a Python 3.12.1 64-bit file.

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

Ans- `import re`

```
def capitalwordsspaces(str1):
    return re.sub(r"(\w) ([A-Z])", r"\1 \2", str1)
print(capitalwordsspaces("RegularExpression1IsAn2ImportantTopic3InPython"))
```



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

Ans- `import re`

```
def text_match(text):
    patterns = '^[a-zA-Z0-9_]*$'
    if re.search(patterns, text):
        return 'Found a match!'
    else:
        return('Not matched!')
```

```
print(text_match("I don't want anything to drink."))
print(text_match("One_Dog_Barks_Loudly_1"))
```


The screenshot shows a Python IDE with a file named `ayush1996.py` open. The code defines a function `text_match(text)` that uses a regular expression `^[a-zA-Z0-9_]*$` to check if a string contains only alphanumeric characters and underscores. It prints the result for two test cases: "I don't want anything to drink." and "One_Dog_Barks_Loudly_1". The terminal output shows "Not matched!" for the first string and "Found a match!" for the second. The Windows taskbar at the bottom shows the date as 4/28/2024 and the time as 8:29 PM.

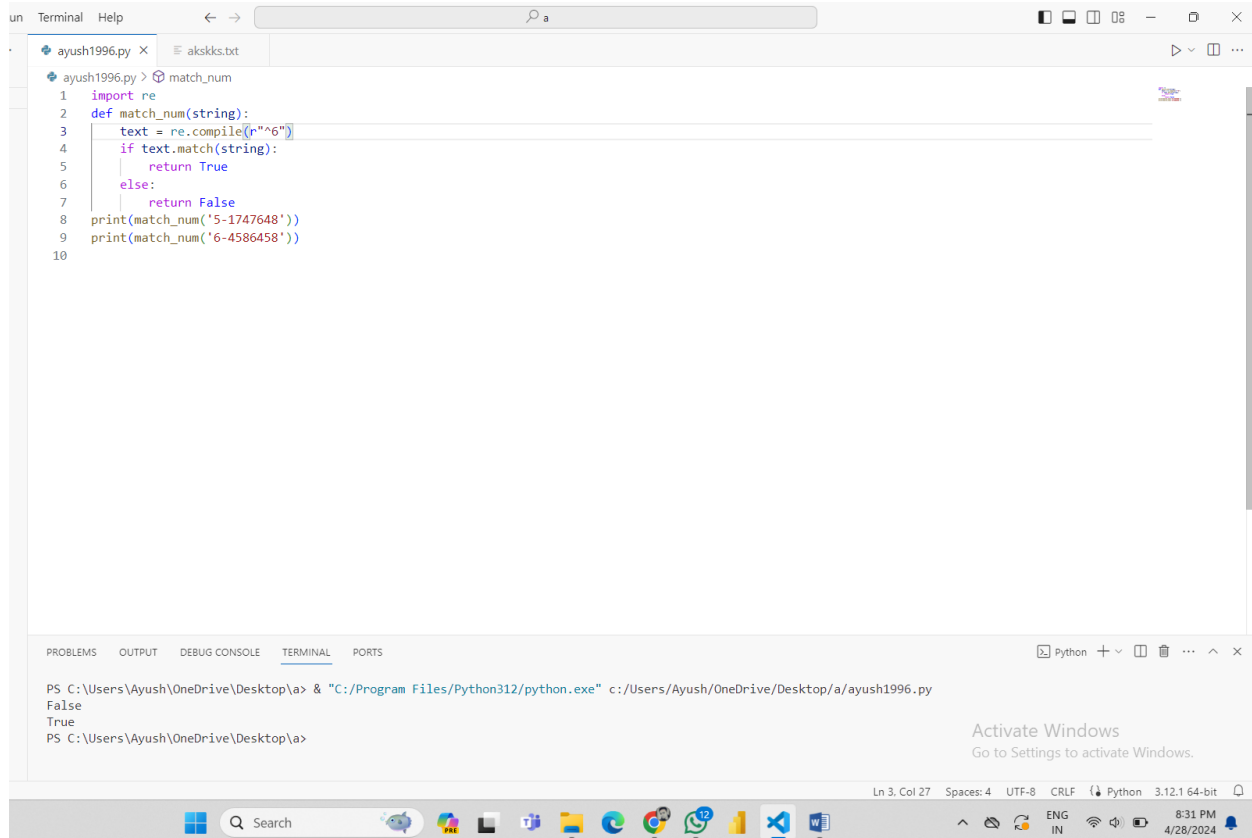
```
1 import re
2 def text_match(text):
3     patterns = '^[a-zA-Z0-9_]*$'
4     if re.search(patterns, text):
5         return 'Found a match!'
6     else:
7         return('Not matched!')
8
9 print(text_match("I don't want anything to drink.))
10 print(text_match("One_Dog_Barks_Loudly_1"))
11
```

PS C:\Users\Ayush\OneDrive\Desktop\a> & "C:/Program Files/Python312/python.exe" c:/Users/Ayush/OneDrive/Desktop/a/ayush1996.py
Not matched!
Found a match!
PS C:\Users\Ayush\OneDrive\Desktop\a>

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

Ans- `import re`

```
def match_num(string):
    text = re.compile(r"^6")
    if text.match(string):
        return True
    else:
        return False
print(match_num('5-1747648'))
print(match_num('6-4586458'))
```



The screenshot shows a Visual Studio Code window with a Python file named `ayush1996.py` open. The code defines a function `match_num` that uses a regular expression to check if a string matches the pattern `^6`. The function returns `True` if the string starts with '6' and `False` otherwise. The script then prints the results of `match_num` for the strings `'5-1747648'` and `'6-4586458'`.

```
1 import re
2 def match_num(string):
3     text = re.compile(r"^6")
4     if text.match(string):
5         return True
6     else:
7         return False
8 print(match_num('5-1747648'))
9 print(match_num('6-4586458'))
10
```

The terminal output shows the execution of the script, resulting in `False` and `True` for the two test cases.

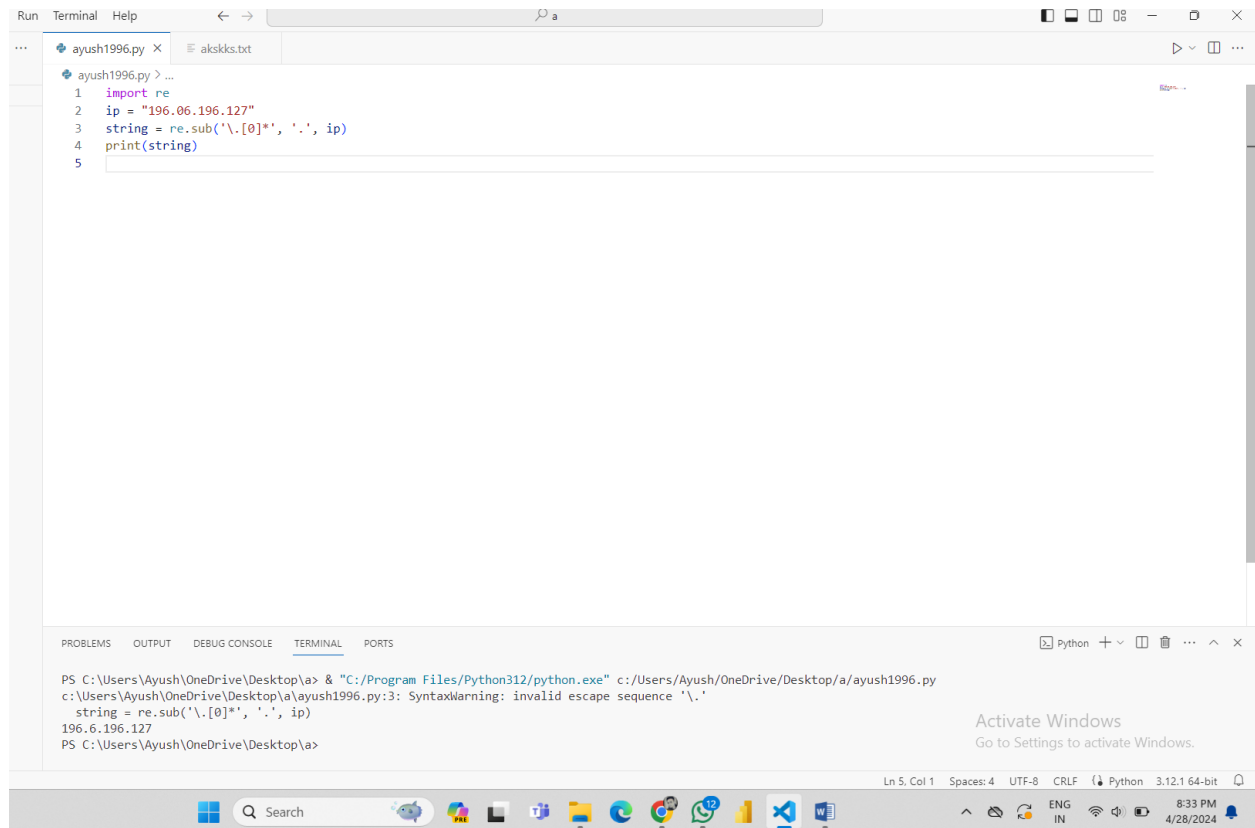
```
PS C:\Users\Ayush\OneDrive\Desktop\ayush1996.py & "C:/Program Files/Python312/python.exe" c:/Users/Ayush/OneDrive/Desktop/a/ayush1996.py
False
True
PS C:\Users\Ayush\OneDrive\Desktop\ayush1996.py
```

The status bar at the bottom indicates the current file is `ayush1996.py`, line 3, column 27, with 4 spaces, UTF-8 encoding, and CRLF line endings. The system tray shows the time as 8:31 PM on 4/28/2024.

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

Ans- `import re`

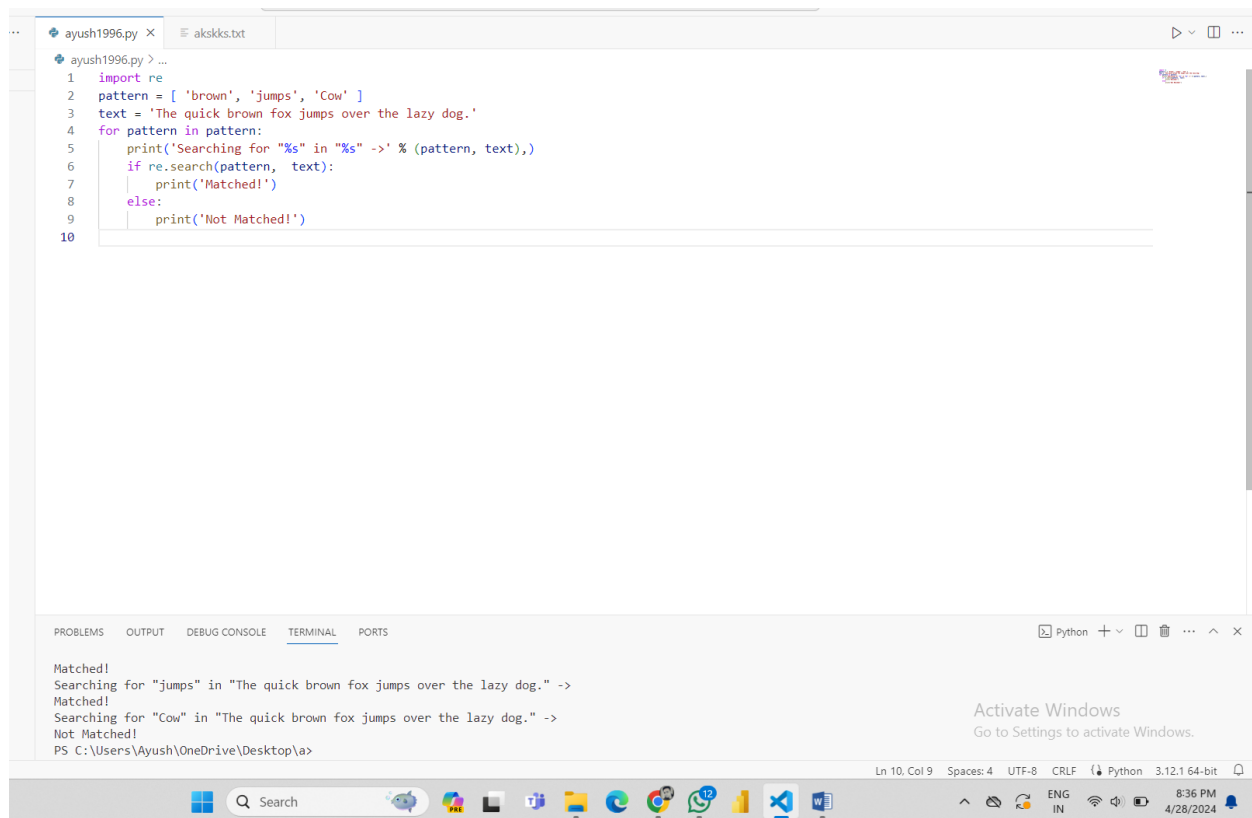
```
ip = "196.06.196.127"
string = re.sub('\.[0]*', '.', ip)
print(string)
```



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

Ans- `import re`

```
pattern = [ 'brown', 'jumps', 'Cow' ]
text = 'The quick brown fox jumps over the lazy dog.'
for pattern in pattern:
    print('Searching for "%s" in "%s" ->' % (pattern, text),)
    if re.search(pattern, text):
        print('Matched!')
    else:
        print('Not Matched!')
```



The screenshot shows a Python IDE with a file named `ayush1996.py` open. The code in the editor is as follows:

```
1 import re
2 pattern = [ 'brown', 'jumps', 'Cow' ]
3 text = 'The quick brown fox jumps over the lazy dog.'
4 for pattern in pattern:
5     print('Searching for "%s" in "%s" ->' % (pattern, text),)
6     if re.search(pattern, text):
7         print('Matched!')
8     else:
9         print('Not Matched!')
```

The terminal output at the bottom shows the execution results:

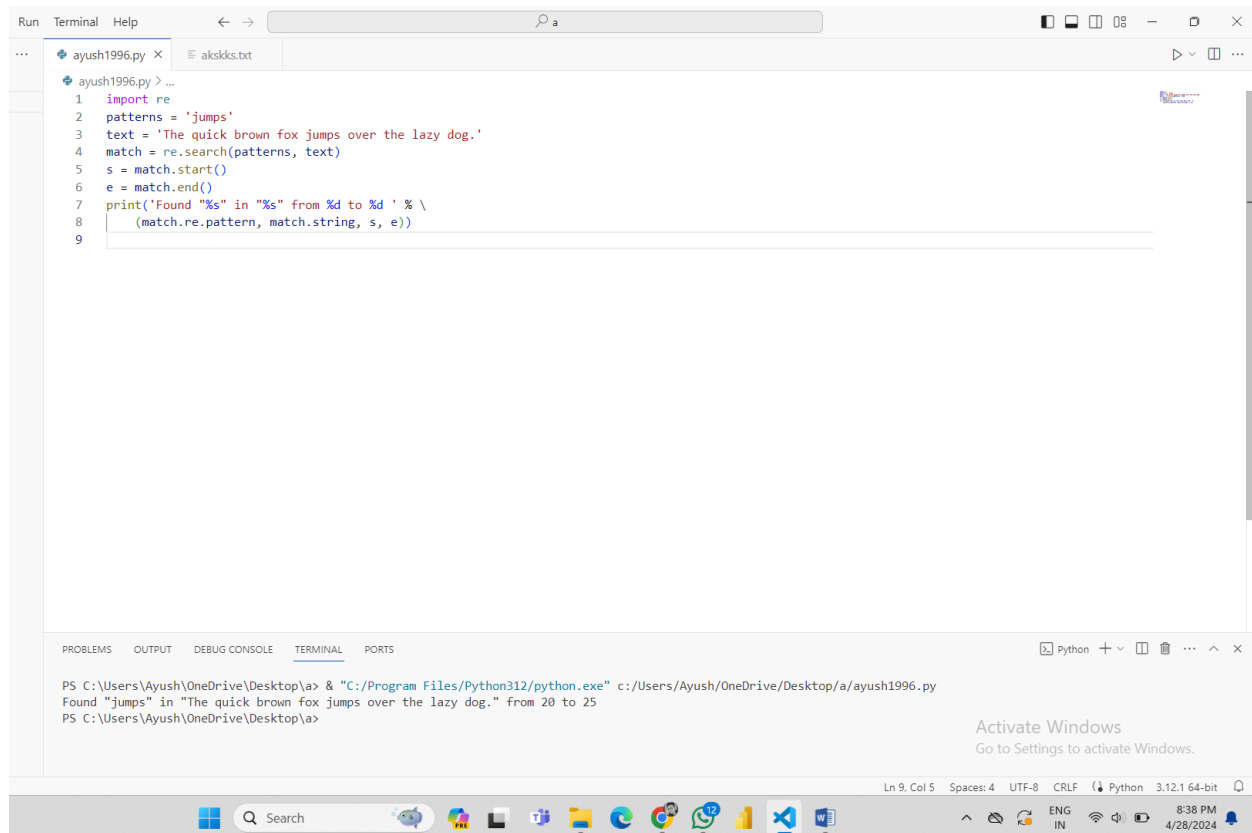
```
Matched!
Searching for "jumps" in "The quick brown fox jumps over the lazy dog." ->
Matched!
Searching for "Cow" in "The quick brown fox jumps over the lazy dog." ->
Not Matched!
PS C:\Users\Ayush\OneDrive\Desktop\>
```

The status bar at the bottom indicates the current line and column (Ln 10, Col 9), encoding (UTF-8), and the Python interpreter version (3.12.1 64-bit).

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

Ans- `import re`

```
patterns = 'jumps'
text = 'The quick brown fox jumps over the lazy dog.'
match = re.search(patterns, text)
s = match.start()
e = match.end()
print('Found "%s" in "%s" from %d to %d ' % \
      (match.re.pattern, match.string, s, e))
```



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

Ans- `import re`

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

The screenshot shows a Python IDE with a file named `ayush1996.py` open. The code in the editor is as follows:

```
1 import re
2 text = 'Python exercise, PHP exercises, C# exercise'
3 patterns = 'exercises'
4 for match in re.findall(patterns, text):
5     print('Found "%s"' % match)
6
```

The terminal output at the bottom shows the command to run the script and the resulting output:

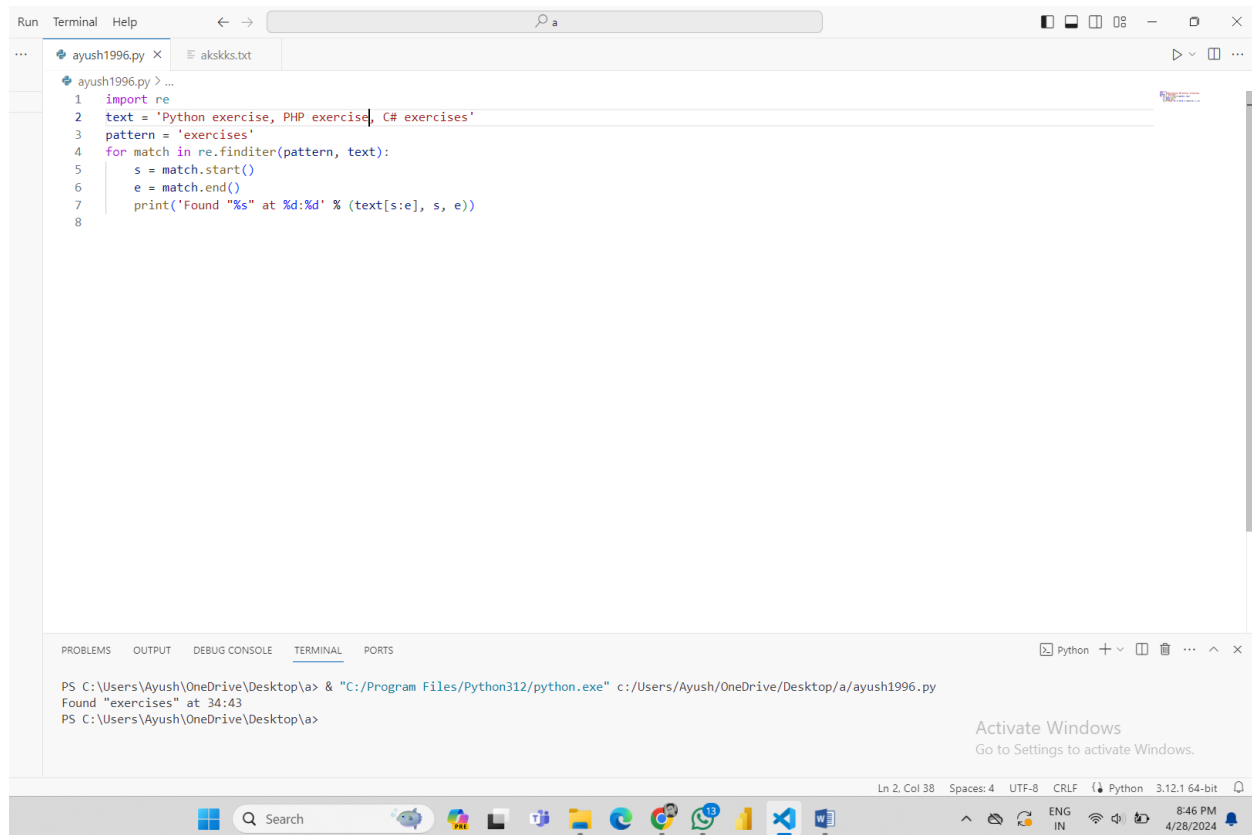
```
PS C:\Users\Ayush\OneDrive\Desktop\> & "C:/Program Files/Python312/python.exe" c:/Users/Ayush/OneDrive/Desktop/a/ayush1996.py
Found "exercises"
PS C:\Users\Ayush\OneDrive\Desktop\>
```

The status bar at the bottom indicates the current line and column (Ln 2, Col 52), encoding (UTF-8), line endings (CRLF), and the Python interpreter path (Python 3.12.1 64-bit).

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

Ans- `import re`

```
text = 'Python exercise, PHP exercise, 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))
```



The screenshot shows a Python IDE with a file named `ayush1996.py` open. The code defines a regular expression pattern `'exercises'` and searches for it in the text `'Python exercise, PHP exercise, C# exercises'`. The output shows the match `'exercises'` at index 34. The terminal window below the editor shows the command `python c:/Users/Ayush/OneDrive/Desktop/a/ayush1996.py` and the output `Found "exercises" at 34:43`. The Windows taskbar at the bottom shows the time as 8:46 PM on 4/28/2024.

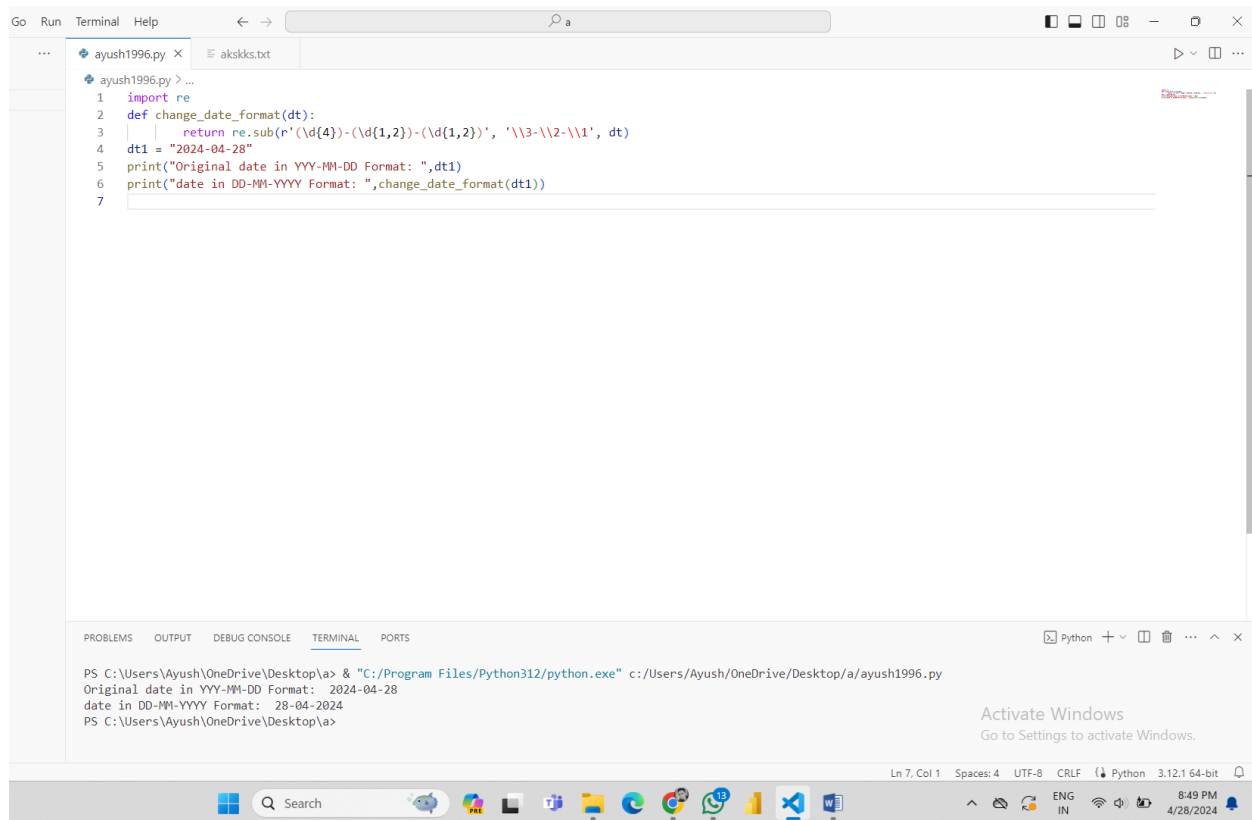
```
1 import re
2 text = 'Python exercise, PHP exercise, C# exercises'
3 pattern = 'exercises'
4 for match in re.finditer(pattern, text):
5     s = match.start()
6     e = match.end()
7     print('Found "%s" at %d:%d' % (text[s:e], s, e))
8
```

PS C:\Users\Ayush\OneDrive\Desktop\a> & "C:/Program Files/Python312/python.exe" c:/Users/Ayush/OneDrive/Desktop/a/ayush1996.py
Found "exercises" at 34:43
PS C:\Users\Ayush\OneDrive\Desktop\a>

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

Ans- `import re`

```
def change_date_format(dt):  
    return re.sub(r'(\d{4})-(\d{1,2})-(\d{1,2})', '\\3-\\2-\\1', dt)  
dt1 = "2024-04-28"  
print("Original date in YYYY-MM-DD Format: ",dt1)  
print("date in DD-MM-YYYY Format: ",change_date_format(dt1))
```



The screenshot shows a Visual Studio Code editor window with a Python file named `ayush1996.py` open. The code defines a function `change_date_format(dt)` that uses a regular expression to reformat a date string. The function takes a date string in 'YYYY-MM-DD' format and returns it in 'DD-MM-YYYY' format. The script then prints the original date and the formatted date.

```
1 import re
2 def change_date_format(dt):
3     return re.sub(r'(\d{4})-(\d{1,2})-(\d{1,2})', '\3-\2-\1', dt)
4 dt1 = "2024-04-28"
5 print("Original date in YYYY-MM-DD Format: ",dt1)
6 print("date in DD-MM-YYYY Format: ",change_date_format(dt1))
7
```

The terminal output shows the execution of the script:

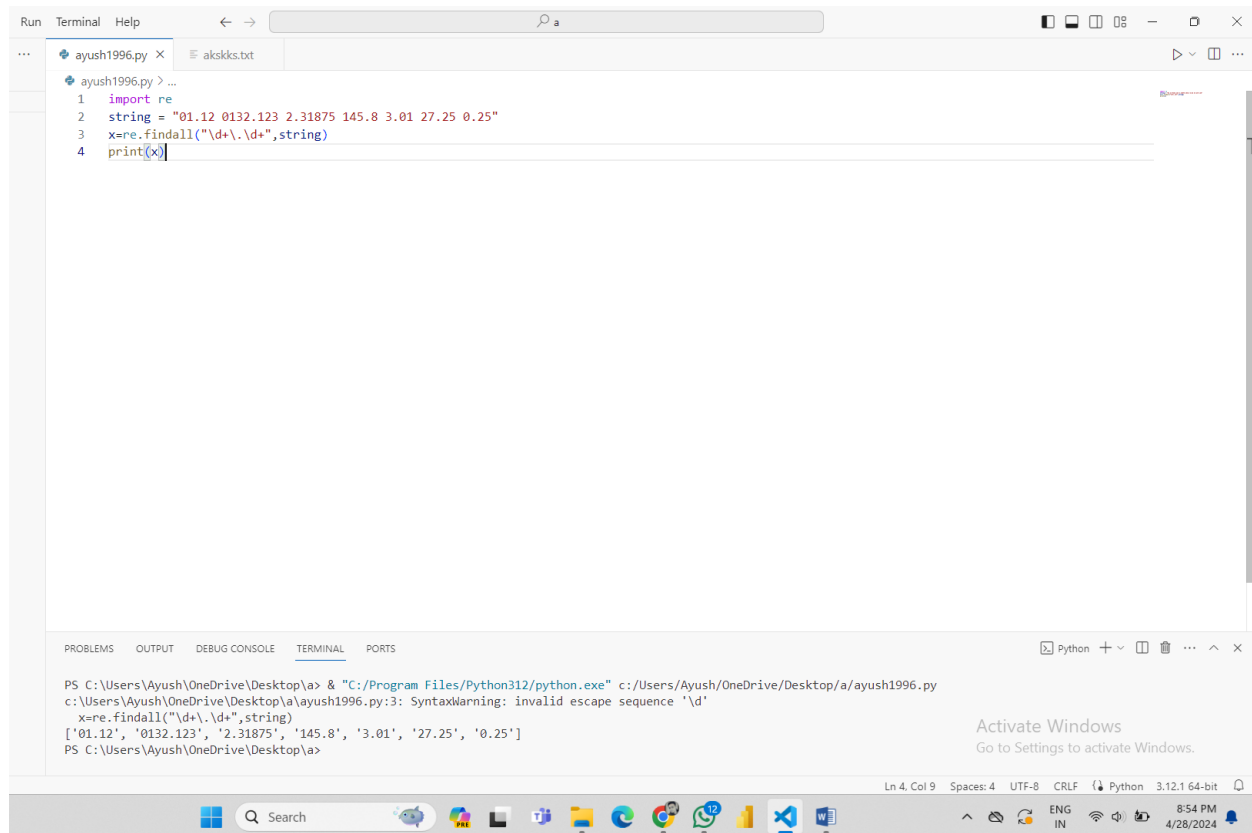
```
PS C:\Users\Ayush\OneDrive\Desktop\a> & "C:/Program Files/Python312/python.exe" c:/Users/Ayush/OneDrive/Desktop/a/ayush1996.py
Original date in YYYY-MM-DD Format: 2024-04-28
date in DD-MM-YYYY Format: 28-04-2024
PS C:\Users\Ayush\OneDrive\Desktop\a>
```

An "Activate Windows" watermark is visible in the bottom right corner of the editor window.

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.

Ans- `import re`

```
string = "01.12 0132.123 2.31875 145.8 3.01 27.25 0.25"
x=re.findall("\d+\.\d+",string)
print(x)
```

```
ayush1996.py > ...
1 import re
2 string = "01.12 0132.123 2.31875 145.8 3.01 27.25 0.25"
3 x=re.findall("\d+\.\d+",string)
4 print(x)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python + Python 3.12.1 64-bit

PS C:\Users\Ayush\OneDrive\Desktop\a> & "C:/Program Files/Python312/python.exe" c:/Users/Ayush/OneDrive/Desktop/a/ayush1996.py
c:\Users\Ayush\OneDrive\Desktop\a\ayush1996.py:3: SyntaxWarning: invalid escape sequence '\d'
x=re.findall("\d+\.\d+",string)
['01.12', '0132.123', '2.31875', '145.8', '3.01', '27.25', '0.25']
PS C:\Users\Ayush\OneDrive\Desktop\a>

Ln 4, Col 9 Spaces: 4 UTF-8 CRLF Python 3.12.1 64-bit

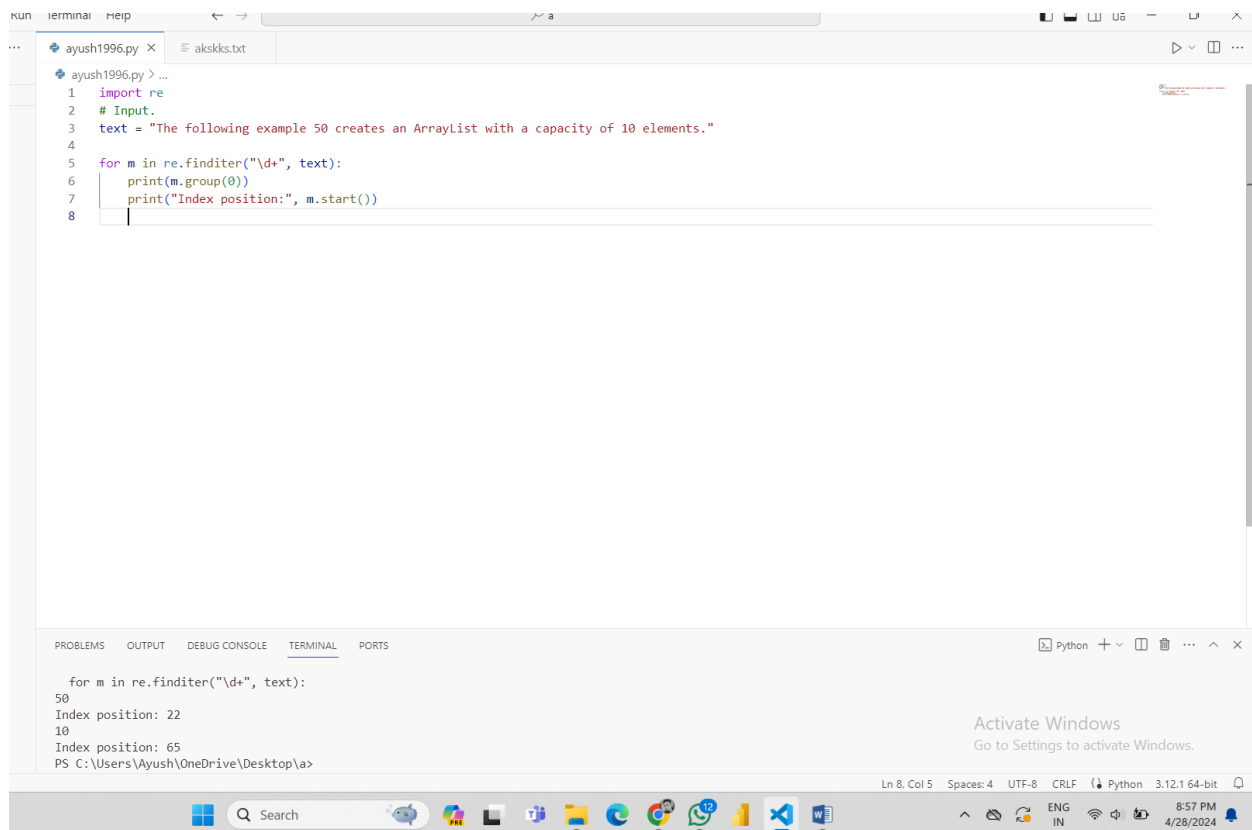
Question 21- Write a Python program to separate and print the numbers and their position of a given string.

Ans- `import re`

`# Input.`

`text = "The following example 50 creates an ArrayList with a capacity of 10 elements."`

```
for m in re.finditer("\d+", text):
    print(m.group(0))
    print("Index position:", m.start())
```



The screenshot shows a Visual Studio Code editor window with a Python file named `ayush1996.py` open. The code in the editor is as follows:

```
1 import re
2 # Input.
3 text = "The following example 50 creates an ArrayList with a capacity of 10 elements."
4
5 for m in re.finditer("\d+", text):
6     print(m.group(0))
7     print("Index position:", m.start())
8
```

The terminal at the bottom shows the output of the script:

```
50
Index position: 22
10
Index position: 65
PS C:\Users\Ayush\OneDrive\Desktop\>a>
```

The status bar at the bottom indicates the current line and column as `Ln 8, Col 5`, and the file encoding as `UTF-8`. The system tray shows the time as `8:57 PM 4/28/2024`.

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

Ans- `import re`

`# Extract all numeric values from the string.`

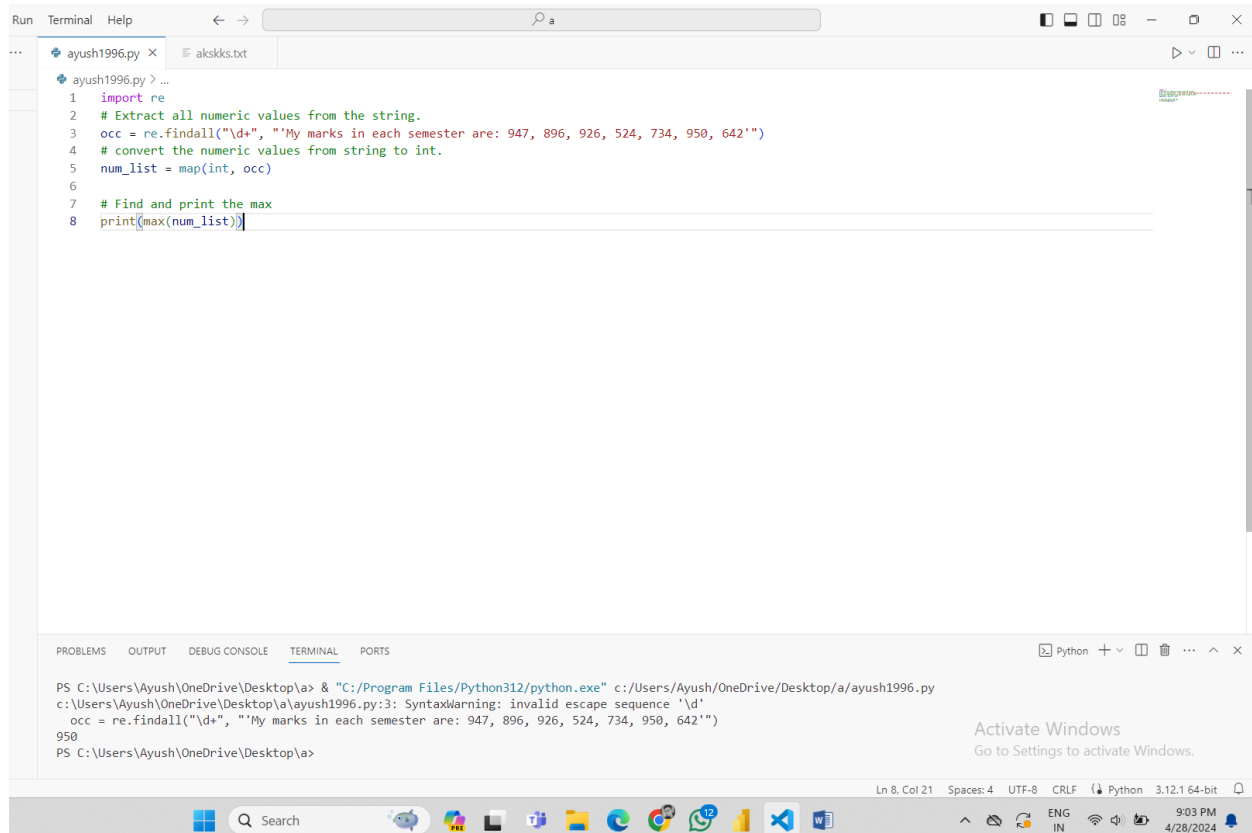
`occ = re.findall("\d+", "'My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'")`

`# convert the numeric values from string to int.`

`num_list = map(int, occ)`

`# Find and print the max`

`print(max(num_list))`



The screenshot shows a Windows IDE with a Python script named `ayush1996.py` and its execution output in the terminal. The script uses `re.findall` to extract numeric values from a string and then prints the maximum value. The terminal output shows a syntax warning for an invalid escape sequence and the final output of 950.

```
1 import re
2 # Extract all numeric values from the string.
3 occ = re.findall("\d+", "My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'")
4 # convert the numeric values from string to int.
5 num_list = map(int, occ)
6
7 # Find and print the max
8 print(max(num_list))
```

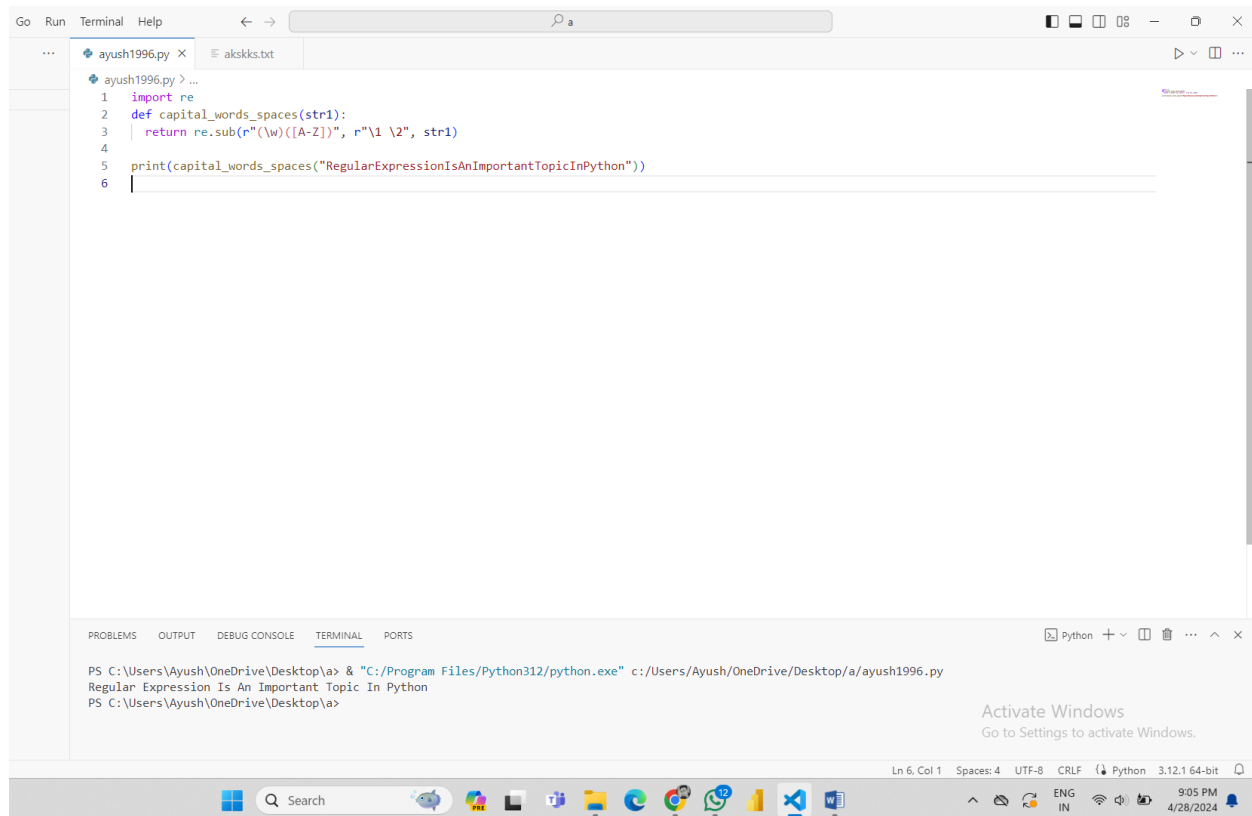
```
PS C:\Users\Ayush\OneDrive\Desktop\ax> & "C:/Program Files/Python312/python.exe" c:/Users/Ayush/OneDrive/Desktop/a/ayush1996.py
c:\Users\Ayush\OneDrive\Desktop\ax\ayush1996.py:3: SyntaxWarning: invalid escape sequence '\d'
  occ = re.findall("\d+", "My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'")
950
PS C:\Users\Ayush\OneDrive\Desktop\ax>
```

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

Ans- `import re`

```
def capital_words_spaces(str1):
    return re.sub(r"(\w)([A-Z])", r"\1 \2", str1)
```

```
print(capital_words_spaces("RegularExpressionIsAnImportantTopicInPython"))
```



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

Ans- `import re`

```
def text_match(text):
    patterns = '[A-Z]+[a-z]+$'
    if re.search(patterns, text):
        return 'Found a match!'
    else:
        return('Not matched!')

print(text_match("Ayush"))
print(text_match("tyagi"))
```

