

# PYTHON ASSIGNMENT - 3

## Q1. Triple Quotes

Code:-

```
paragraph = """
```

Python is a high-level, interpreted programming language known for its simplicity and readability.

Created by Guido van Rossum and first released in 1991, Python emphasizes code readability with its

notable use of significant whitespace. It supports multiple programming paradigms, including procedural,

object-oriented, and functional programming. Python's extensive standard library and vibrant community

make it a popular choice for web development, data analysis, artificial intelligence, scientific computing,

and more.

```
"""
```

```
print(paragraph)
```

```
word = input("Enter your desired word: ")
```

```
sentence = paragraph.split(".")
```

```
for sentence in sentence:
```

```
    if word in sentence:
```

```
        print(sentence)
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS C:\Users\daksh\Desktop\Python_Practical> python -u "c:\Users\daksh\Desktop\Python_Practical\Labwork-3\q1.py"

Python is a high-level, interpreted programming language known for its simplicity and readability.
Created by Guido van Rossum and first released in 1991, Python emphasizes code readability with its
notable use of significant whitespace. It supports multiple programming paradigms, including procedural,
object-oriented, and functional programming. Python's extensive standard library and vibrant community
make it a popular choice for web development, data analysis, artificial intelligence, scientific computing,
and more.

Enter your desired word: Python

Python is a high-level, interpreted programming language known for its simplicity and readability

Created by Guido van Rossum and first released in 1991, Python emphasizes code readability with its
notable use of significant whitespace
    Python's extensive standard library and vibrant community
make it a popular choice for web development, data analysis, artificial intelligence, scientific computing,
and more
PS C:\Users\daksh\Desktop\Python_Practical> 
```

## Q2.Slicing String

Code:-

```
string = "Hello, World!"
```

```
length = int(len(string)/2)
```

```
print("First half of the string: ")
```

```
print(string[0:length])
```

```
print("Second half of the string: ")
```

```
print(string[length:])
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS C:\Users\daksh\Desktop\Python_Practical> python -u "c:\Users\daksh\De
First half of the string:
Hello,
Second half of the string:
World!
PS C:\Users\daksh\Desktop\Python_Practical> |
```

### Q3. String Concatenation

Code:-

```
def concatenate_strings(str1,str2):
```

```
    if str2 == "":
```

```
        return str1
```

```
    else:
```

```
        return str1 + " " + str2
```

```
string1 =str(input("Enter the first string: "))
```

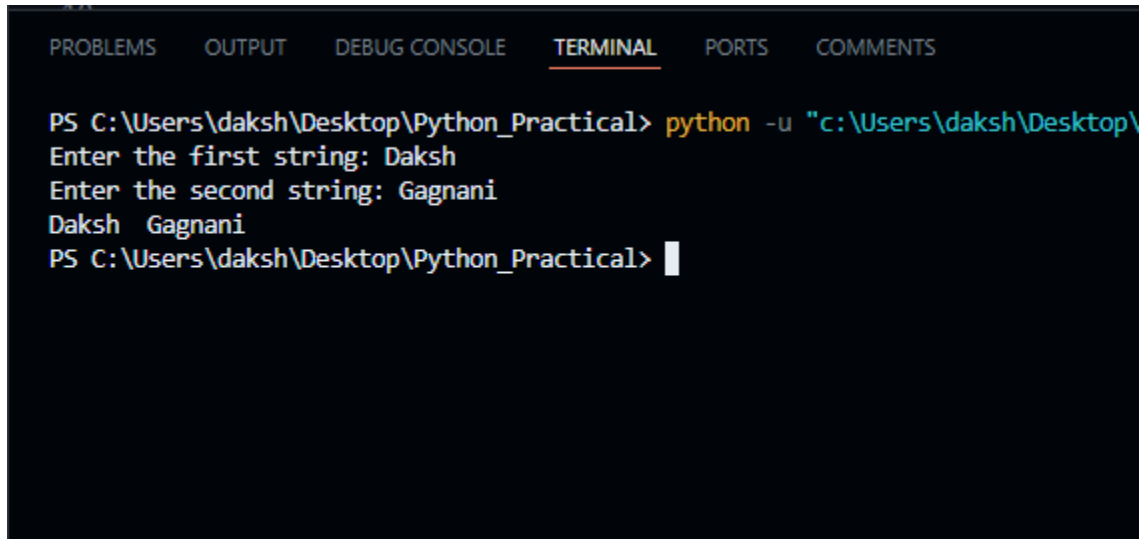
```
string2 =str(input("Enter the second string: "))
```

```
print(concatenate_strings(string1,string2))
```

```
string = "Hello, World!"
```

```
print(string[:5])
```

Output:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS C:\Users\daksh\Desktop\Python_Practical> python -u "c:\Users\daksh\Desktop\Python_Practical\center_string.py"
Enter the first string: Daksh
Enter the second string: Gagnani
Daksh Gagnani
PS C:\Users\daksh\Desktop\Python_Practical>
```

#### 4. String Methods

Code:-

```
def center_string(string):
    return string.center(30)
```

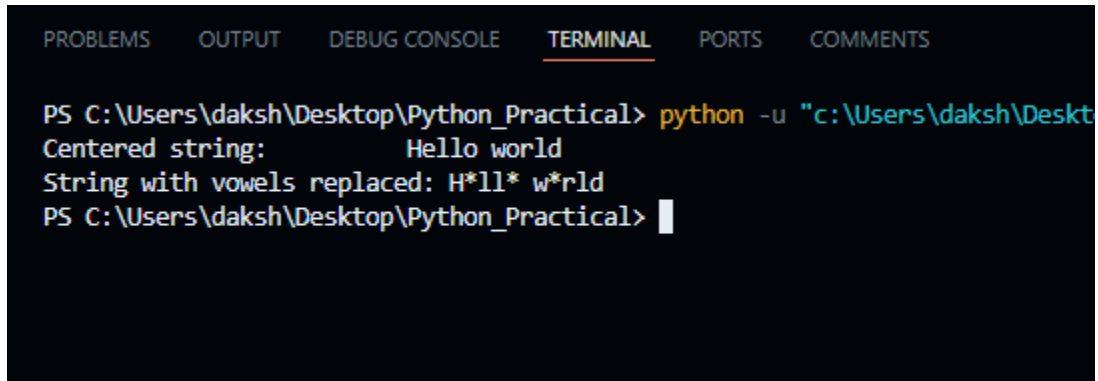
```
def replace_vowel(string):
    vowels = "aeiouAEIOU"
    for i in vowels:
        string = string.replace(i, '*')
    return string
```

```
string = "Hello world"
```

```
print("Centered string:", center_string(string))
```

```
print("String with vowels replaced:", replace_vowel(string))
```

Output:-

A screenshot of a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active and underlined), 'PORTS', and 'COMMENTS'. The terminal shows a command prompt 'PS C:\Users\daksh\Desktop\Python\_Practical>' followed by the command 'python -u "c:\Users\daksh\Desktop\Python\_Practical\code.py"'. The output of the script is displayed: 'Centered string: Hello world' and 'String with vowels replaced: H\*ll\* w\*rld'. The prompt returns to 'PS C:\Users\daksh\Desktop\Python\_Practical>' with a cursor at the end.

```
PS C:\Users\daksh\Desktop\Python_Practical> python -u "c:\Users\daksh\Desktop\Python_Practical\code.py"
Centered string:      Hello world
String with vowels replaced: H*ll* w*rld
PS C:\Users\daksh\Desktop\Python_Practical>
```

## 5. Assignment Operators

Code:-

```
num = int(input("Enter an integer: "))
print("Original number:", num)
num += 5
print("After adding 5:", num)
num -= 5
print("After subtracting 5:", num)
num *= 5
print("After multiplying by 5:", num)
num /= 5
print("After dividing by 5:", num)
num //= 5
print("After floor dividing by 5:", num)
num %= 5
print("After taking the remainder by 5:", num)
```

Output:-

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS C:\Users\daksh\Desktop\Python_Practical> python -u "c:\Users\daksh\Desktop\Python_
Enter an integer: 5
Original number: 5
After adding 5: 10
After subtracting 5: 5
After multiplying by 5: 25
After dividing by 5: 5.0
After floor dividing by 5: 1.0
After taking the remainder by 5: 1.0
PS C:\Users\daksh\Desktop\Python_Practical> |
```

## 6. Password Validator (Combining Concepts)

Code:-

```
def passwordValidator(password):
    if len(password) < 8:
        return False
    if not any(char.isupper() for char in password):
        return False
    if not any(char.islower() for char in password):
        return False
    if not any(char.isdigit() for char in password):
        return False
    if ' ' in password:
        return False
    return True
```

```
password = input("Enter your password: ")
```

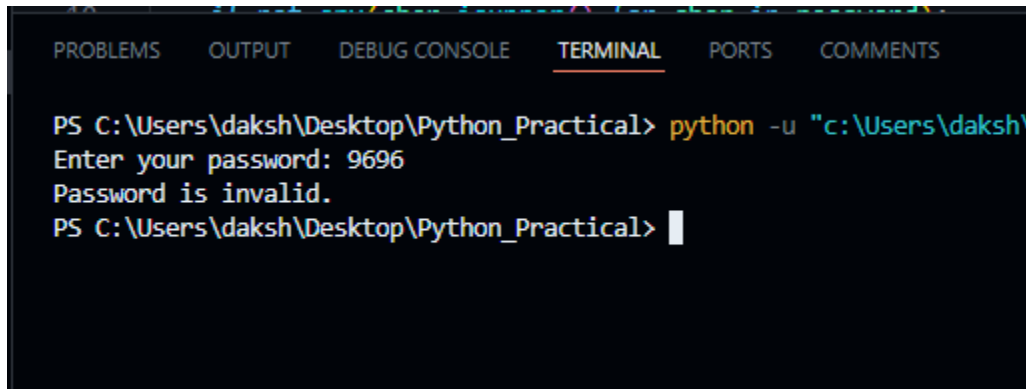
```
if passwordValidator(password):
```

```
    print("Password is valid.")
```

```
else:
```

```
    print("Password is invalid.")
```

Output:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS  
PS C:\Users\daksh\Desktop\Python_Practical> python -u "c:\Users\daksh\  
Enter your password: 9696  
Password is invalid.  
PS C:\Users\daksh\Desktop\Python_Practical> |
```

## 7. Acronym Generator

Code:-

```
string = input("Enter a string: ")
```

```
words = string.split()
```

```
acronym = ".join(word[0].upper() for word in words)
```

```
print("Acronym:", acronym)
```

Output:-

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS C:\Users\daksh\Desktop\Python_Practical> python -u "c:\Users
Enter a string: Daksh Gagnani
Acronym: DG
PS C:\Users\daksh\Desktop\Python_Practical> |
```

## 8. Simple Calculator

Code:-

```
def calculator(num1, num2, operator):
```

```
    match operator:
```

```
        case '+':
```

```
            return num1 + num2
```

```
        case '-':
```

```
            return num1 - num2
```

```
        case '*':
```

```
            return num1 * num2
```

```
        case '/':
```

```
            return num1 / num2
```

```
        case '//':
```

```
            return num1 // num2
```

```
        case '%':
```

```
            return num1 % num2
```

```
        case '**':
```

```
            return num1 ** num2
```



```
case _:  
    return "Invalid operator"
```

```
while True:
```

```
    user_input = input("Enter '0' to exit or press Enter to continue: ")
```

```
    if user_input == '0':
```

```
        break
```

```
    num1 = float(input("Enter the first number: "))
```

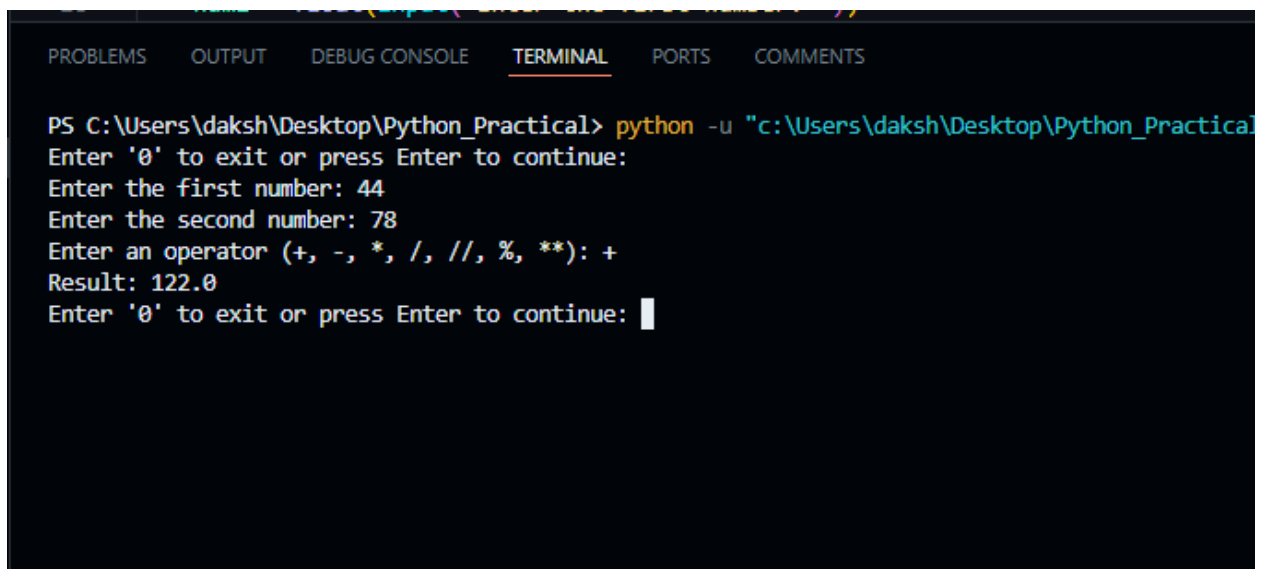
```
    num2 = float(input("Enter the second number: "))
```

```
    operator = input("Enter an operator (+, -, *, /, //, %, **): ")
```

```
    result = calculator(num1, num2, operator)
```

```
    print("Result:", result)
```

Output:-



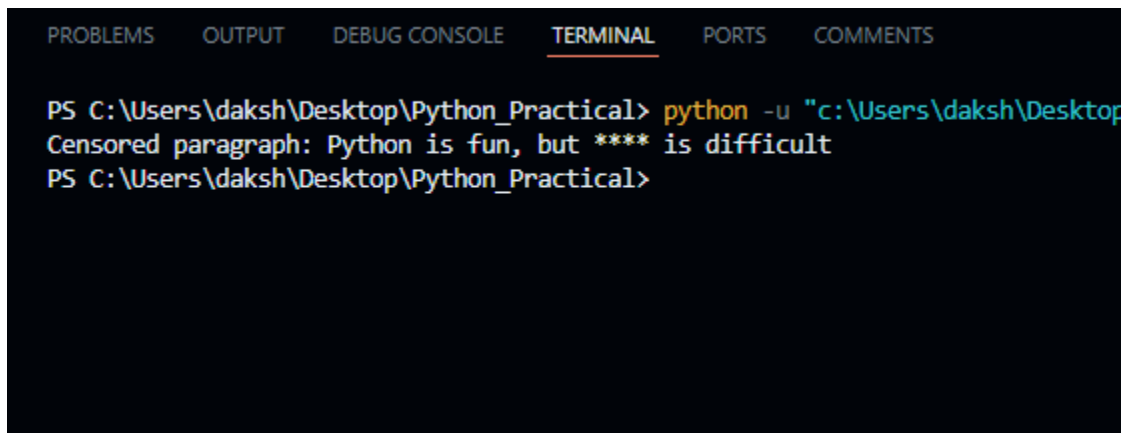
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS  
PS C:\Users\daksh\Desktop\Python_Practical> python -u "c:\Users\daksh\Desktop\Python_Practical  
Enter '0' to exit or press Enter to continue:  
Enter the first number: 44  
Enter the second number: 78  
Enter an operator (+, -, *, /, //, %, **): +  
Result: 122.0  
Enter '0' to exit or press Enter to continue: |
```

## 9. Word Censorship

Code:-

```
def censor_word(paragraph, banned_word):  
    return paragraph.replace(banned_word, "****")  
  
paragraph = "Python is fun, but Java is difficult"  
banned_word = "Java"  
  
censored_paragraph = censor_word(paragraph, banned_word)  
print("Censored paragraph:", censored_paragraph)
```

Output:-



The screenshot shows a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), 'PORTS', and 'COMMENTS'. The terminal content shows a command prompt 'PS C:\Users\daksh\Desktop\Python\_Practical>' followed by the command 'python -u "c:\Users\daksh\Desktop\Python\_Practical\word\_censorship.py"'. The output of the script is displayed on the next line: 'Censored paragraph: Python is fun, but \*\*\*\* is difficult'. The prompt returns to 'PS C:\Users\daksh\Desktop\Python\_Practical>'.

## 10. Anagram Checker

Code:-

```
def are_anagrams(str1, str2):  
    str1 = str1.replace(" ", "").lower()  
    str2 = str2.replace(" ", "").lower()
```

```
str2 = str2.replace(" ", "").lower()
```

```
if sorted(str1) == sorted(str2):
```

```
    print("is anagram")
```

```
else:
```

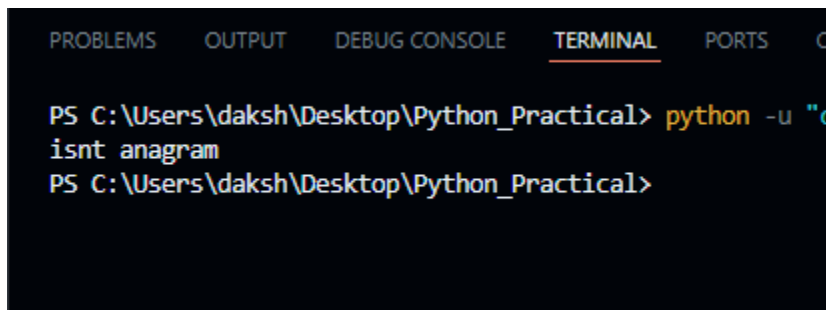
```
    print("isnt anagram")
```

```
str1 = "hi"
```

```
str2 = "ii"
```

```
are_anagrams(str1, str2)
```

Output:-



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  CO  
PS C:\Users\daksh\Desktop\Python_Practical> python -u "c  
isnt anagram  
PS C:\Users\daksh\Desktop\Python_Practical>
```

## 11. Find the Longest Word in a Sentence

Code:-

```
def find_longest_word(sentence):
```

```
    words = sentence.split()
```

```
    longest_word = max(words, key=len)
```

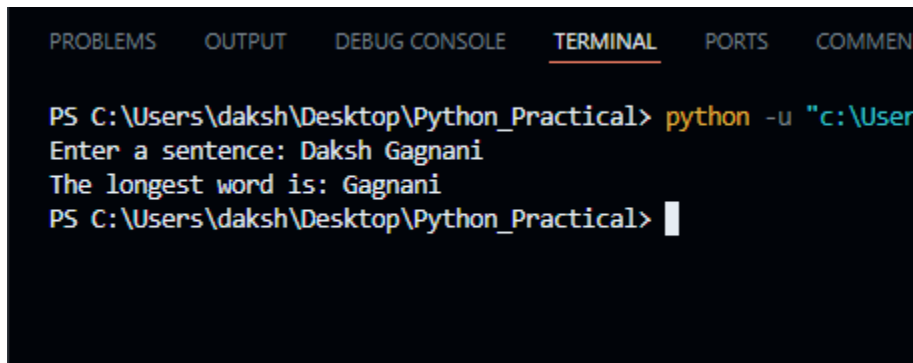
```
return longest_word
```

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

```
longest_word = find_longest_word(sentence)
```

```
print("The longest word is:", longest_word)
```

Output:-



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS C:\Users\daksh\Desktop\Python_Practical> python -u "c:\User
Enter a sentence: Daksh Gagnani
The longest word is: Gagnani
PS C:\Users\daksh\Desktop\Python_Practical>
```

## 12. Sort Words in a Sentence Alphabetically

Code:-

```
def sort_words(sentence):
```

```
    words = sentence.split()
```

```
    sorted_words = sorted(words)
```

```
    return ' '.join(sorted_words)
```

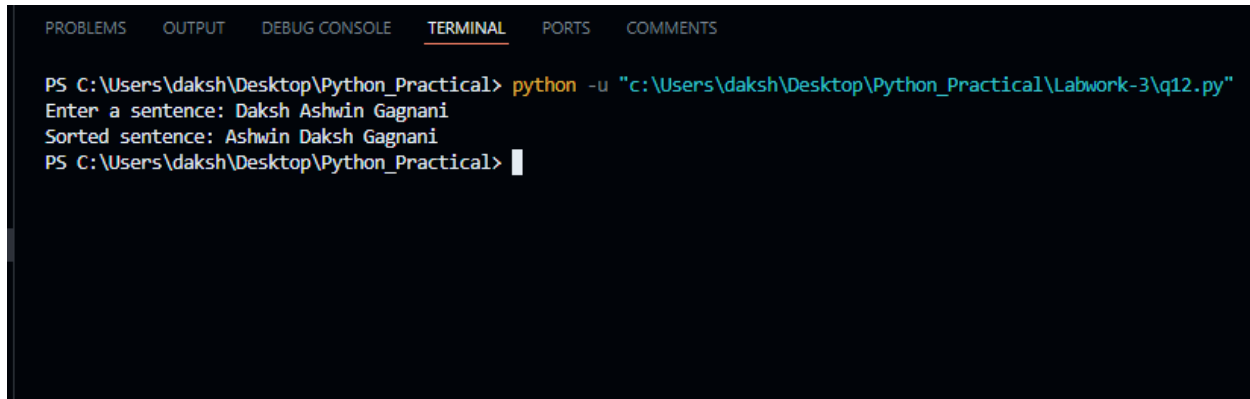
```
# Example usage
```

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

```
sorted_sentence = sort_words(sentence)
```

```
print("Sorted sentence:", sorted_sentence)
```

Output:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS  
PS C:\Users\daksh\Desktop\Python_Practical> python -u "c:\Users\daksh\Desktop\Python_Practical\Labwork-3\q12.py"  
Enter a sentence: Daksh Ashwin Gagnani  
Sorted sentence: Ashwin Daksh Gagnani  
PS C:\Users\daksh\Desktop\Python_Practical>
```

### 13. Count Words in a Sentence

Code:-

```
def count_words(sentence):
```

```
    words = sentence.split()
```

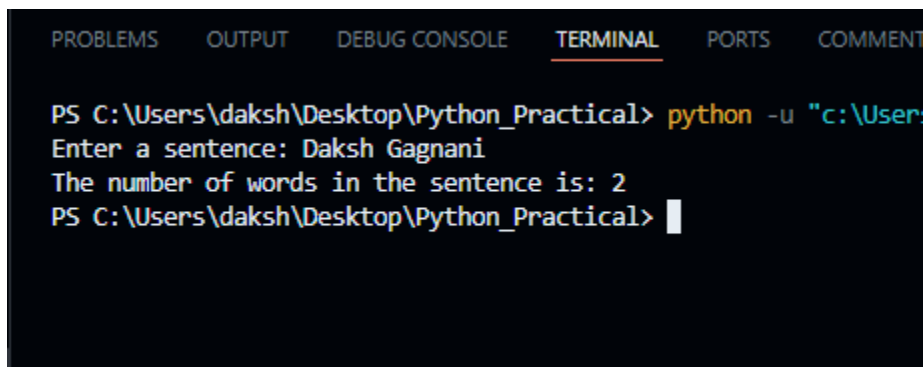
```
    return len(words)
```

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

```
word_count = count_words(sentence)
```

```
print("The number of words in the sentence is:", word_count)
```

Output:-



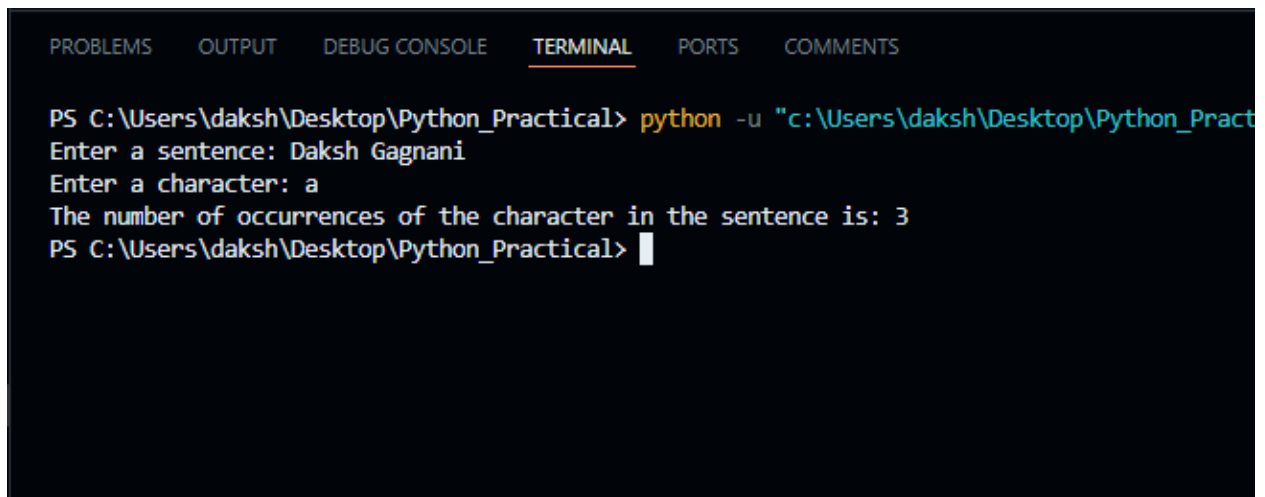
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS  
PS C:\Users\daksh\Desktop\Python_Practical> python -u "c:\Users\daksh\Desktop\Python_Practical\Labwork-3\q13.py"  
Enter a sentence: Daksh Gagnani  
The number of words in the sentence is: 2  
PS C:\Users\daksh\Desktop\Python_Practical>
```

#### 14. Count the Occurrences of a Given Character

Code:-

```
def count_characters(sentence, character):  
    return sentence.count(character)  
  
sentence = input("Enter a sentence: ")  
character = input("Enter a character: ")  
character_count = count_characters(sentence, character)  
print("The number of occurrences of the character in the sentence is:", character_count)
```

Output:-



The screenshot shows a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), 'PORTS', and 'COMMENTS'. The terminal content shows the following sequence of commands and outputs:

```
PS C:\Users\daksh\Desktop\Python_Practical> python -u "c:\Users\daksh\Desktop\Python_Pract  
Enter a sentence: Daksh Gagnani  
Enter a character: a  
The number of occurrences of the character in the sentence is: 3  
PS C:\Users\daksh\Desktop\Python_Practical> █
```

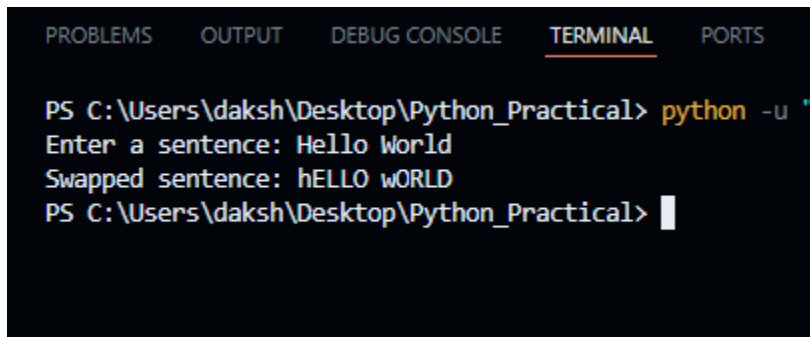
#### 15. Swap Case

Code:-

```
def swap_case(sentence):  
    return sentence.swapcase()
```

```
sentence = input("Enter a sentence: ")  
swapped_sentence = swap_case(sentence)  
print("Swapped sentence:", swapped_sentence)
```

Output:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS C  
PS C:\Users\daksh\Desktop\Python_Practical> python -u "c  
Enter a sentence: Hello World  
Swapped sentence: hELLO wORLD  
PS C:\Users\daksh\Desktop\Python_Practical> █
```

## 16. Mask Credit Card Number

Code:-

```
def mask_credit_card_number(credit_card_number):  
    masked_credit_card_number = '*' * 12 + credit_card_number[-4:]  
    return masked_credit_card_number  
  
credit_card_number = input("Enter a credit card number: ")  
masked_credit_card_number = mask_credit_card_number(credit_card_number)  
print("Masked credit card number:", masked_credit_card_number)
```

Output:-

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

COMMENTS

```
PS C:\Users\daksh\Desktop\Python_Practical> python -u "c:\Users\d
Enter a credit card number: 1234823912
Masked credit card number: *****3912
PS C:\Users\daksh\Desktop\Python_Practical> |
```