1. Write a Python program to calculate the length of a string.

ANS:

```
string = input("Enter a string: ")
length = len(string)
print("Length of the string:", length)
```

2. Write a Python program to count the number of characters (character frequency) in a string.
Sample String : google.com'
Expected Result : {'o': 3, 'g': 2, '.': 1, 'e': 1, 'l': 1, 'm': 1, 'c': 1}

ANS:

```
string = 'google.com'
char_frequency = {}
for char in string:
    if char in char_frequency:
        char_frequency[char] += 1
    else:
        char_frequency[char] = 1
print("Character frequency:", char_frequency)
```

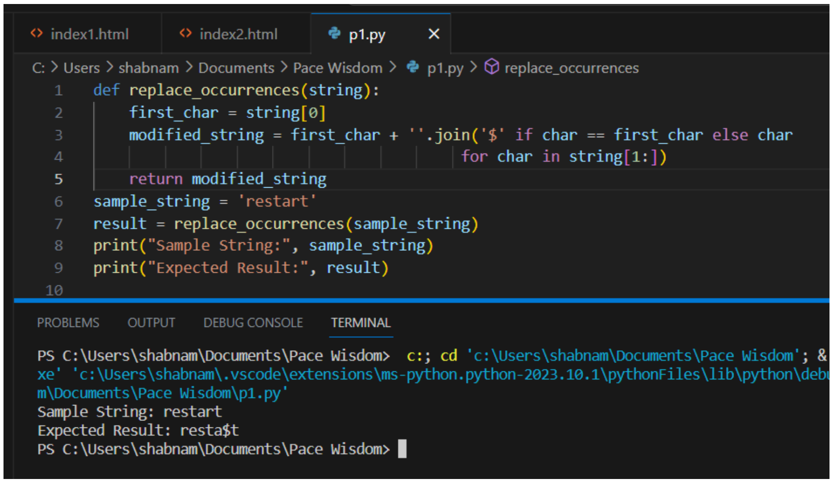
3. Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead of the empty string.
Sample String : 'thisisniceone'
Expected Result : 'thne'"
Sample String : 'ab'
Expected Result : 'abab'
Sample String : 'f'
Expected Result : Empty String

ANS:

4. Write a Python program to get a string from a given string where all occurrences of its first char have been changed to '$', except the first char itself.
Sample String : 'restart'
Expected Result : 'resta$t'

ANS:

```
def replace_occurrences(string):
   first_char = string[0]
   modified_string = first_char + ''.join('$' if char == first_char else char
                                            for char in string[1:])
   return modified_string
sample_string = 'restart'
result = replace_occurrences(sample_string)
print("Sample String:", sample_string)
print("Expected Result:", result)
```


index1.html
index2.html
p1.py
C: > Users > shabnam > Documents > Pace Wisdom > p1.py > replace_occurrences
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

```
def replace_occurrences(string):
    first_char = string[0]
    modified_string = first_char + ''.join('$' if char == first_char else char
                                           for char in string[1:])
    return modified_string
sample_string = 'restart'
result = replace_occurrences(sample_string)
print("Sample String:", sample_string)
print("Expected Result:", result)
```

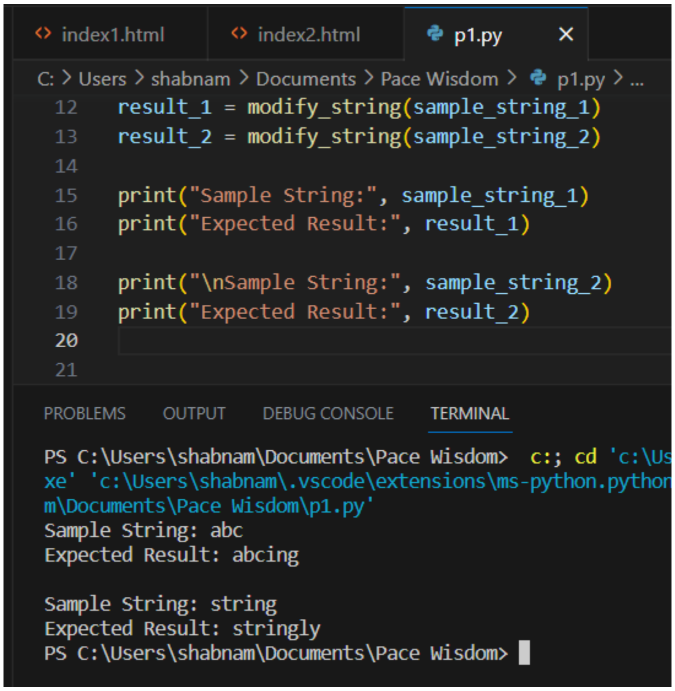
```
PS C:\Users\shabnam\Documents\Pace Wisdom>  c:; cd 'c:\Users\shabnam\Documents\Pace Wisdom'; &
xe' 'c:\Users\shabnam\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debu
m\Documents\Pace Wisdom\p1.py'
Sample String: restart
Expected Result: resta$t
PS C:\Users\shabnam\Documents\Pace Wisdom>
```

5. Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string.
Sample String : 'abc', 'xyz'
Expected Result : 'xyc abz'

6. Write a Python program to add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged.
Sample String : 'abc'
Expected Result : 'abcing'
Sample String : 'string'
Expected Result : 'stringly'

ANS:

```
def modify_string(string):

    if len(string) < 3:
        return string
    elif string[-3:] == 'ing':
        return string + 'ly'
    else:
        return string + 'ing'

sample_string_1 = 'abc'
sample_string_2 = 'string'

result_1 = modify_string(sample_string_1)
result_2 = modify_string(sample_string_2)

print("Sample String:", sample_string_1)
print("Expected Result:", result_1)

print("\nSample String:", sample_string_2)
print("Expected Result:", result_2)
```



```
result_1 = modify_string(sample_string_1)
result_2 = modify_string(sample_string_2)

print("Sample String:", sample_string_1)
print("Expected Result:", result_1)

print("\nSample String:", sample_string_2)
print("Expected Result:", result_2)

```

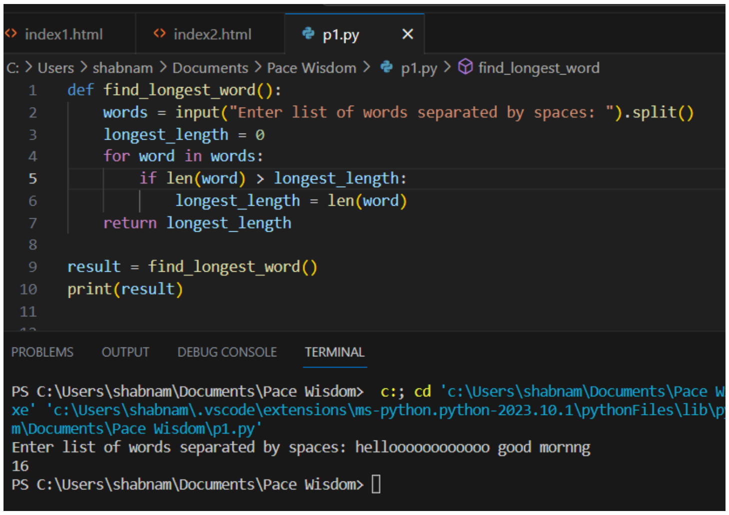
```
PS C:\Users\shabnam\Documents\Pace Wisdom>  c:; cd 'c:\Us
xe' 'c:\Users\shabnam\.vscode\extensions\ms-python.python
m\Documents\Pace Wisdom\p1.py'
Sample String: abc
Expected Result: abcing

Sample String: string
Expected Result: stringly
PS C:\Users\shabnam\Documents\Pace Wisdom> 
```

7. Write a Python program to find the first appearance of the substring 'not' and 'poor' from a given string, if 'not' follows the 'poor', replace the whole 'not'...'poor' substring with 'good'. Return the resulting string.
Sample String : 'The lyrics is not that poor!'
'The lyrics is poor!'
Expected Result : 'The lyrics is good!'
'The lyrics is poor!'

8. Write a Python function that takes a list of words and returns the length of the longest one.

ANS:

```
def find_longest_word():
    words = input("Enter list of words separated by spaces: ").split()
    longest_length = 0
    for word in words:
        if len(word) > longest_length:
            longest_length = len(word)
    return longest_length

result = find_longest_word()
print(result)
```


index1.html
index2.html
p1.py
C: > Users > shabnam > Documents > Pace Wisdom > p1.py > find_longest_word
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

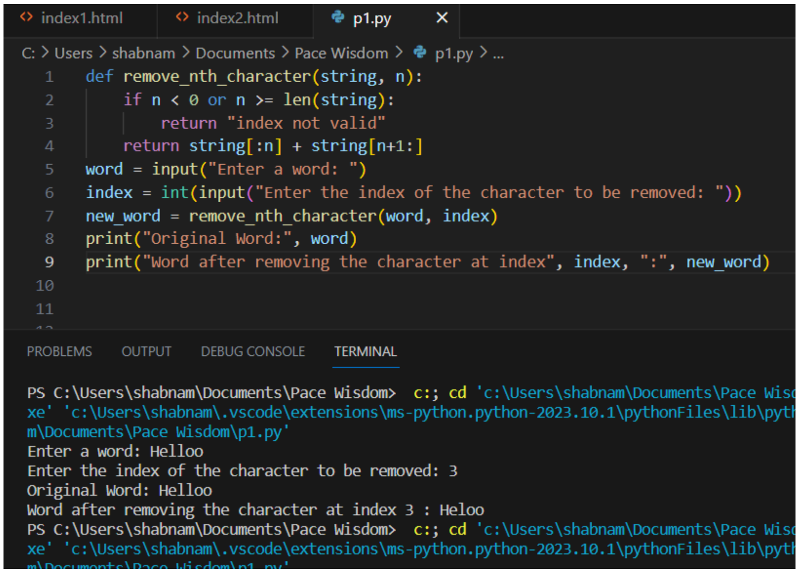
```
def find_longest_word():
    words = input("Enter list of words separated by spaces: ").split()
    longest_length = 0
    for word in words:
        if len(word) > longest_length:
            longest_length = len(word)
    return longest_length

result = find_longest_word()
print(result)
```

```
PS C:\Users\shabnam\Documents\Pace Wisdom>  c:; cd 'c:\Users\shabnam\Documents\Pace W
xe' 'c:\Users\shabnam\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\p
m\Documents\Pace Wisdom\p1.py'
Enter list of words separated by spaces: helloooooooooooo good mornng
16
PS C:\Users\shabnam\Documents\Pace Wisdom> 
```

9. Write a Python program to remove the nth index character from a nonempty string.

ANS:

```
def remove_nth_character(string, n):
    if n < 0 or n >= len(string):
        return "index not valid"
    return string[:n] + string[n+1:]
word = input("Enter a word: ")
index = int(input("Enter the index of the character to be removed: "))
new_word = remove_nth_character(word, index)
print("Original Word:", word)
print("Word after removing the character at index", index, ":", new_word)
```



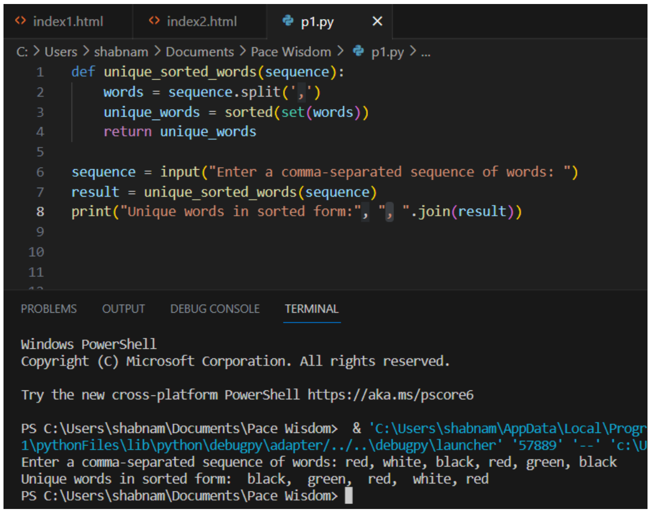
```
def remove_nth_character(string, n):
    if n < 0 or n >= len(string):
        return "index not valid"
    return string[:n] + string[n+1:]
word = input("Enter a word: ")
index = int(input("Enter the index of the character to be removed: "))
new_word = remove_nth_character(word, index)
print("Original Word:", word)
print("Word after removing the character at index", index, ":", new_word)
```

```
PS C:\Users\shabnam\Documents\Pace Wisdom>  c:; cd 'c:\Users\shabnam\Documents\Pace Wisd
xe' 'c:\Users\shabnam\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\pyth
m\Documents\Pace Wisdom\p1.py'
Enter a word: Helloo
Enter the index of the character to be removed: 3
Original Word: Helloo
Word after removing the character at index 3 : Heloo
PS C:\Users\shabnam\Documents\Pace Wisdom>  c:; cd 'c:\Users\shabnam\Documents\Pace Wisd
xe' 'c:\Users\shabnam\.vscode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\pyth
m\Documents\Pace Wisdom\p1.py'
```

10. Write a Python program that accepts a comma separated sequence of words as input and prints the unique words in sorted form (alphanumerically).
Sample Words : red, white, black, red, green, black
Expected Result : black, green, red, white

ANS:

```
def unique_sorted_words(sequence):
    words = sequence.split(',')
    unique_words = sorted(set(words))
    return unique_words

sequence = input("Enter a comma-separated sequence of words: ")
result = unique_sorted_words(sequence)
print("Unique words in sorted form:", ", ".join(result))
```



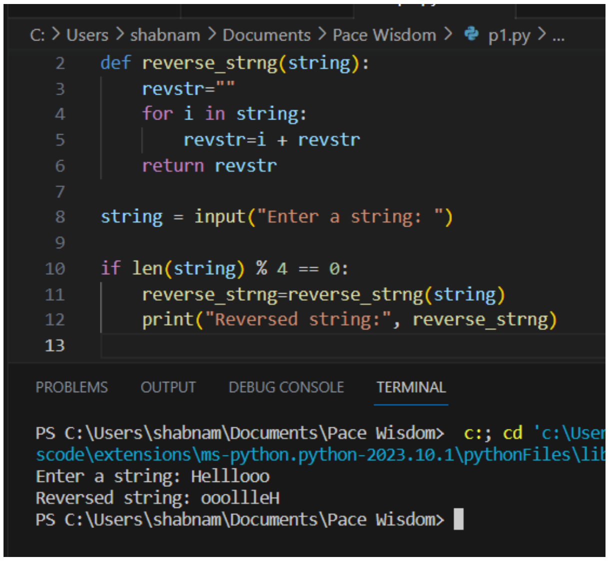
```
def unique_sorted_words(sequence):
    words = sequence.split(',')
    unique_words = sorted(set(words))
    return unique_words

sequence = input("Enter a comma-separated sequence of words: ")
result = unique_sorted_words(sequence)
print("Unique words in sorted form:", ", ".join(result))
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\shabnam\Documents\Pace Wisdom>  & 'C:\Users\shabnam\AppData\Local\Progr
1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '57889' '--' 'c:\U
Enter a comma-separated sequence of words: red, white, black, red, green, black
Unique words in sorted form:  black,  green,  red,  white, red
PS C:\Users\shabnam\Documents\Pace Wisdom>
```

11. Write a Python function to reverses a string if it's length is a multiple of 4.

ANS:

```
def reverse_strng(string):
    revstr=""
    for i in string:
        revstr=i + revstr
    return revstr

string = input("Enter a string: ")

if len(string) % 4 == 0:
    reverse_strng=reverse_strng(string)
    print("Reversed string:", reverse_strng)
```


C: > Users > shabnam > Documents > Pace Wisdom > p1.py > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

```
def reverse_strng(string):
    revstr=""
    for i in string:
        revstr=i + revstr
    return revstr

string = input("Enter a string: ")

if len(string) % 4 == 0:
    reverse_strng=reverse_strng(string)
    print("Reversed string:", reverse_strng)
```

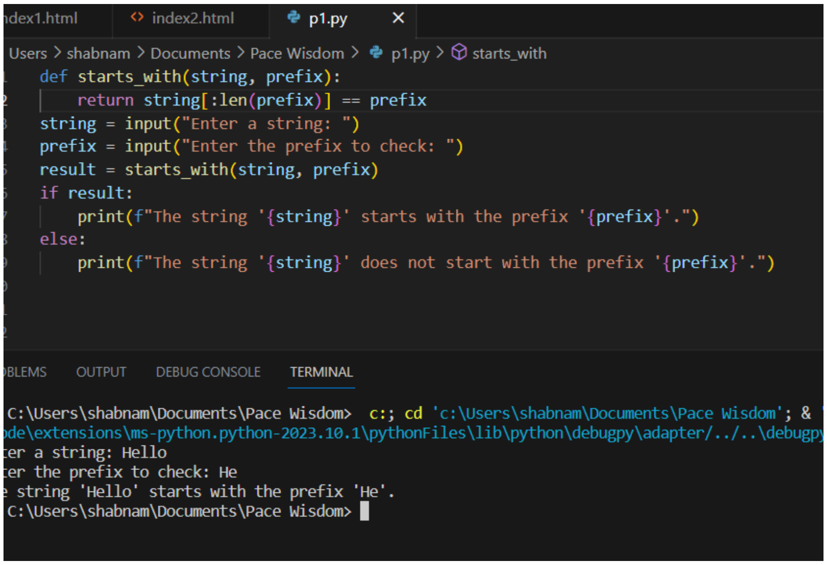
```
PS C:\Users\shabnam\Documents\Pace Wisdom>  c:; cd 'c:\User
scode\extensions\ms-python.python-2023.10.1\pythonFiles\lib
Enter a string: Helllooo
Reversed string: ooollleH
PS C:\Users\shabnam\Documents\Pace Wisdom> 
```

12. Write a Python function to convert a given string to all uppercase if it contains at least 2 uppercase characters in the first 4 characters.

13. Write a Python program to check whether a string starts with specified characters.

ANS:

```
def starts_with(string, prefix):
    return string[:len(prefix)] == prefix
string = input("Enter a string: ")
prefix = input("Enter the prefix to check: ")
result = starts_with(string, prefix)
if result:
    print(f"The string '{string}' starts with the prefix '{prefix}'.")
else:
    print(f"The string '{string}' does not start with the prefix '{prefix}'.")
```



ndex1.html
index2.html
p1.py
Users > shabnam > Documents > Pace Wisdom > p1.py > starts_with

```
def starts_with(string, prefix):
    return string[:len(prefix)] == prefix
string = input("Enter a string: ")
prefix = input("Enter the prefix to check: ")
result = starts_with(string, prefix)
if result:
    print(f"The string '{string}' starts with the prefix '{prefix}'.")
else:
    print(f"The string '{string}' does not start with the prefix '{prefix}'.")
```

BLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

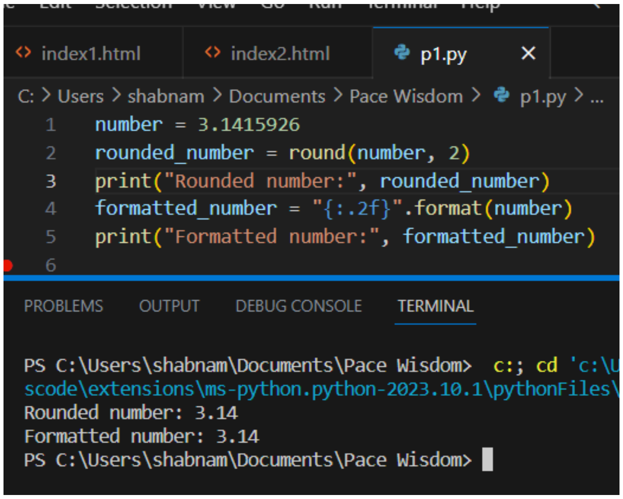
```
C:\Users\shabnam\Documents\Pace Wisdom>  c:; cd 'c:\Users\shabnam\Documents\Pace Wisdom'; & '
ode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy
ter a string: Hello
ter the prefix to check: He
 string 'Hello' starts with the prefix 'He'.
C:\Users\shabnam\Documents\Pace Wisdom>
```

14. Write a Python program to print the following floating numbers upto 2 decimal places.
3.1415926

ANS:

```
number = 3.1415926
```

```
rounded_number = round(number, 2)
print("Rounded number:", rounded_number)
formatted_number = "{:.2f}".format(number)
print("Formatted number:", formatted_number)
```


Edit Selection View Go Run Terminal Help
index1.html
index2.html
p1.py
C: > Users > shabnam > Documents > Pace Wisdom > p1.py > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

```
number = 3.1415926
rounded_number = round(number, 2)
print("Rounded number:", rounded_number)
formatted_number = "{:.2f}".format(number)
print("Formatted number:", formatted_number)
```

```
PS C:\Users\shabnam\Documents\Pace Wisdom>  c:; cd 'c:\U
scode\extensions\ms-python.python-2023.10.1\pythonFiles\
Rounded number: 3.14
Formatted number: 3.14
PS C:\Users\shabnam\Documents\Pace Wisdom> 
```

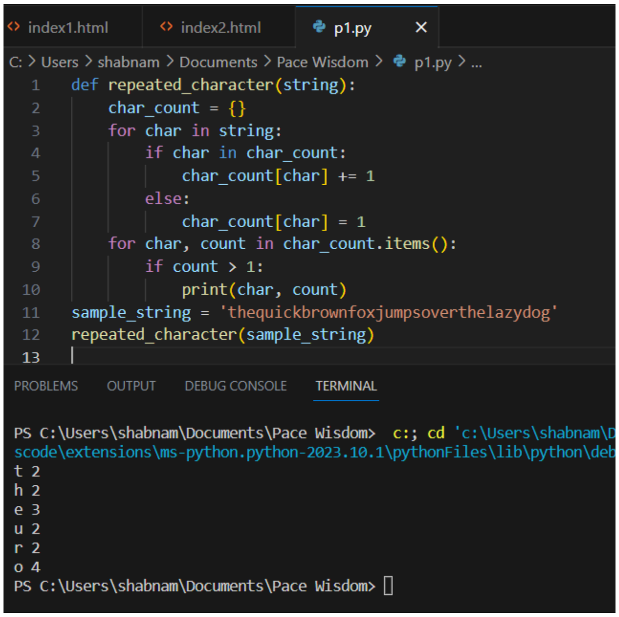
15. Write a Python program to count repeated characters in a string.
Sample string: 'thequickbrownfoxjumpsoverthelazydog'
Expected output :

```
o 4
e 3
u 2
h 2
r 2
t 2
```

ANS:

```
def repeated_character(string):
    char_count = {}
    for char in string:
        if char in char_count:
            char_count[char] += 1
        else:
            char_count[char] = 1
    for char, count in char_count.items():
        if count > 1:
            print(char, count)
sample_string = 'thequickbrownfoxjumpsoverthelazydog'
```

repeated_character(sample_string)



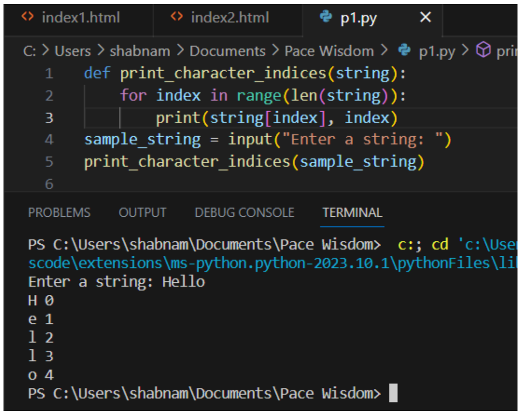
```
def repeated_character(string):
    char_count = {}
    for char in string:
        if char in char_count:
            char_count[char] += 1
        else:
            char_count[char] = 1
    for char, count in char_count.items():
        if count > 1:
            print(char, count)
sample_string = 'thequickbrownfoxjumpsoverthelazydog'
repeated_character(sample_string)
```

```
PS C:\Users\shabnam\Documents\Pace Wisdom>  c:; cd 'c:\Users\shabnam\D
scode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\deb
t 2
h 2
e 3
u 2
r 2
o 4
PS C:\Users\shabnam\Documents\Pace Wisdom> 
```

16. Write a Python program to print the index of the character in a string.

ANS:

```
def print_character_indices(string):
    for index in range(len(string)):
        print(string[index], index)
sample_string = input("Enter a string: ")
print_character_indices(sample_string)
```



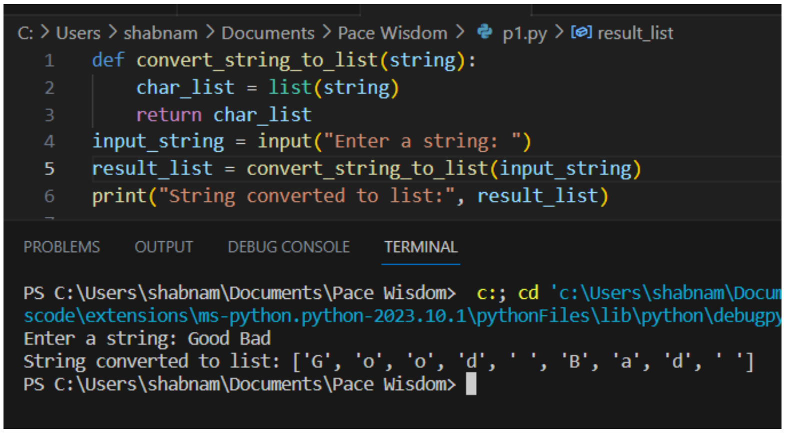
```
def print_character_indices(string):
    for index in range(len(string)):
        print(string[index], index)
sample_string = input("Enter a string: ")
print_character_indices(sample_string)
```

```
PS C:\Users\shabnam\Documents\Pace Wisdom>  c:; cd 'c:\Use
scode\extensions\ms-python.python-2023.10.1\pythonFiles\li
Enter a string: Hello
H 0
e 1
l 2
l 3
o 4
PS C:\Users\shabnam\Documents\Pace Wisdom> 
```

17. Write a Python program to convert a string in a list.

ANS:

```
def convert_string_to_list(string):
    char_list = list(string)
    return char_list
input_string = input("Enter a string: ")
result_list = convert_string_to_list(input_string)
print("String converted to list:", result_list)
```


C: > Users > shabnam > Documents > Pace Wisdom > p1.py > result_list
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

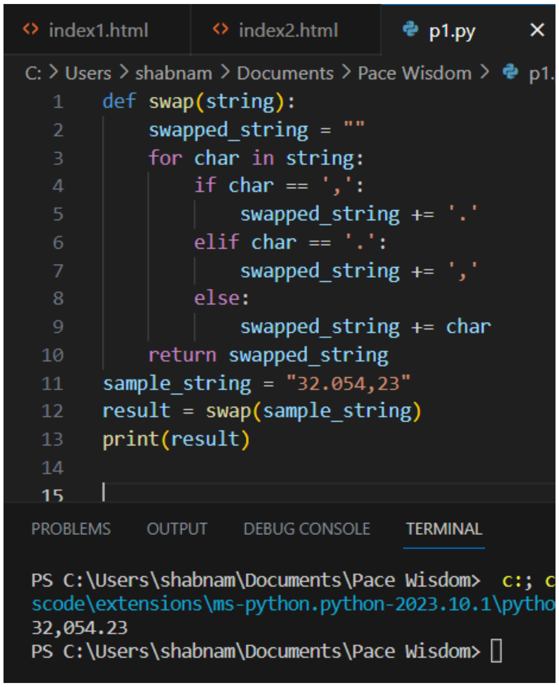
```
def convert_string_to_list(string):
    char_list = list(string)
    return char_list
input_string = input("Enter a string: ")
result_list = convert_string_to_list(input_string)
print("String converted to list:", result_list)
```

```
PS C:\Users\shabnam\Documents\Pace Wisdom>  c:; cd 'c:\Users\shabnam\Docum
scode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\debugpy
Enter a string: Good Bad
String converted to list: ['G', 'o', 'o', 'd', ' ', 'B', 'a', 'd', ' ']
PS C:\Users\shabnam\Documents\Pace Wisdom> 
```

18. Write a Python program to swap comma and dot in a string.
Sample string: "32.054,23"
Expected Output: "32,054.23"

ANS:

```
def swap(string):
    swapped_string = ""
    for char in string:
        if char == ',':
            swapped_string += '.'
        elif char == '.':
            swapped_string += ','
        else:
            swapped_string += char
    return swapped_string
sample_string = "32.054,23"
result = swap(sample_string)
print(result)
```



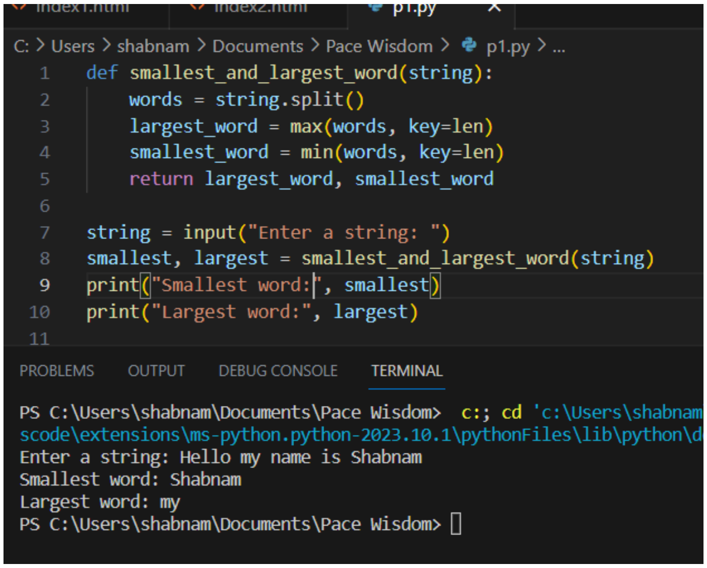
```python
def swap(string):
    swapped_string = ""
    for char in string:
        if char == ',':
            swapped_string += '.'
        elif char == '.':
            swapped_string += ','
        else:
            swapped_string += char
    return swapped_string
sample_string = "32.054,23"
result = swap(sample_string)
print(result)

```

```
PS C:\Users\shabnam\Documents\Pace Wisdom>  c:; c
scode\extensions\ms-python.python-2023.10.1\pytho
32,054.23
PS C:\Users\shabnam\Documents\Pace Wisdom>
```

19. Write a Python program to find smallest and largest word in a given string.

ANS:

```python
def smallest_and_largest_word(string):
    words = string.split()
    largest_word = max(words, key=len)
    smallest_word = min(words, key=len)
    return largest_word, smallest_word

string = input("Enter a string: ")
smallest, largest = smallest_and_largest_word(string)
print("Smallest word:", smallest)
print("Largest word:", largest)
```



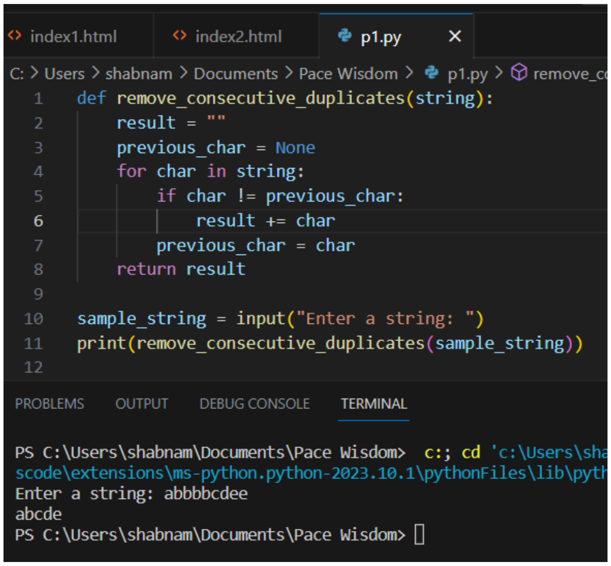
```python
def smallest_and_largest_word(string):
    words = string.split()
    largest_word = max(words, key=len)
    smallest_word = min(words, key=len)
    return largest_word, smallest_word

string = input("Enter a string: ")
smallest, largest = smallest_and_largest_word(string)
print("Smallest word:", smallest)
print("Largest word:", largest)
```

```
PS C:\Users\shabnam\Documents\Pace Wisdom>  c:; cd 'c:\Users\shabnam
scode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\d
Enter a string: Hello my name is Shabnam
Smallest word: Shabnam
Largest word: my
PS C:\Users\shabnam\Documents\Pace Wisdom> 
```

20. Write a Python program to remove all consecutive duplicates of a given string.

ANS:

```python
def remove_consecutive_duplicates(string):
    result = ""
    previous_char = None
    for char in string:
        if char != previous_char:
            result += char
        previous_char = char
    return result

sample_string = input("Enter a string: ")
print(remove_consecutive_duplicates(sample_string))
```



index1.html
index2.html
p1.py
C: > Users > shabnam > Documents > Pace Wisdom > p1.py > remove_c
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

```python
def remove_consecutive_duplicates(string):
    result = ""
    previous_char = None
    for char in string:
        if char != previous_char:
            result += char
        previous_char = char
    return result

sample_string = input("Enter a string: ")
print(remove_consecutive_duplicates(sample_string))
```

```
PS C:\Users\shabnam\Documents\Pace Wisdom>  c:; cd 'c:\Users\sha
scode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\pyth
Enter a string: abbbbcdee
abcde
PS C:\Users\shabnam\Documents\Pace Wisdom> 
```