

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](https://www.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.

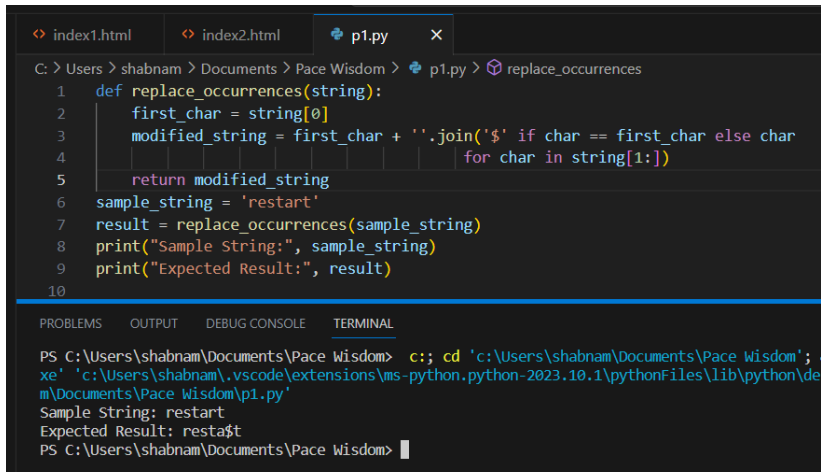
Sample String : 'restart'

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)
```



```
C: > Users > shabnam > Documents > Pace Wisdom > p1.py > replace_occurrences
1 def replace_occurrences(string):
2     first_char = string[0]
3     modified_string = first_char + '.join('$' if char == first_char else char
4                                     for char in string[1:])
5     return modified_string
6 sample_string = 'restart'
7 result = replace_occurrences(sample_string)
8 print("Sample String:", sample_string)
9 print("Expected Result:", result)
10

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
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):
```

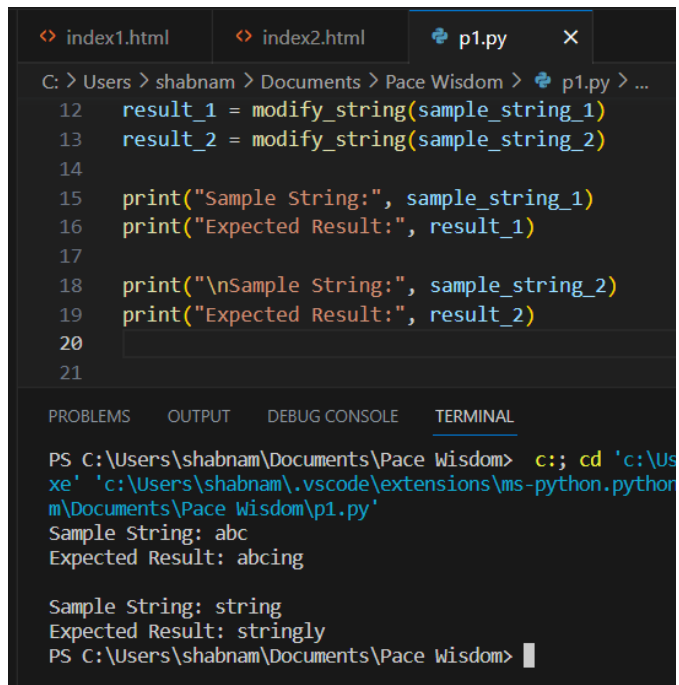
```
    if len(string) < 3:
        return string
    elif string[-3:] == 'ing':
        return string + 'ly'
    else:
        return string + 'ing'
```

```
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)
```



The screenshot shows a VS Code editor with a file named `p1.py` open. The code in the editor is as follows:

```
12 result_1 = modify_string(sample_string_1)
13 result_2 = modify_string(sample_string_2)
14
15 print("Sample String:", sample_string_1)
16 print("Expected Result:", result_1)
17
18 print("\nSample String:", sample_string_2)
19 print("Expected Result:", result_2)
20
21
```

Below the editor, the TERMINAL panel is active, showing the command prompt output:

```
PS C:\Users\shabnam\Documents\Pace Wisdom> c::; cd 'c:\Us
xe' 'c:\Users\shabnam\.vscode\extensions\ms-python.pytho
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!'

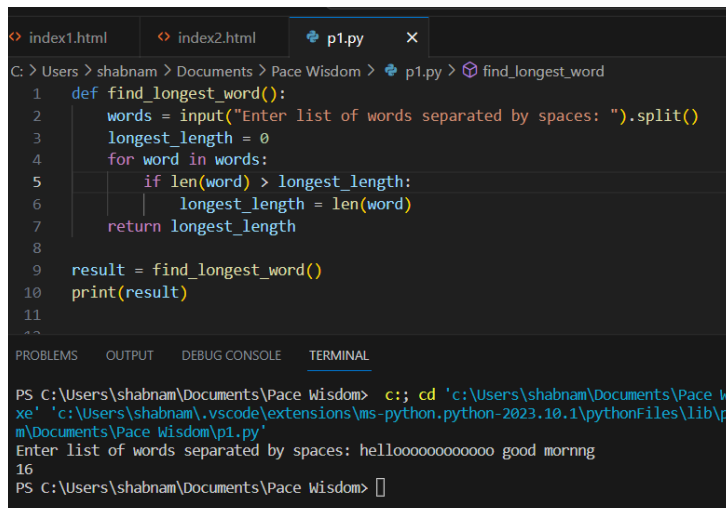
'The lyrics is poor!'

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)
```



The screenshot shows a VS Code editor with a file named `p1.py` open. The code in the editor is as follows:

```
1 def find_longest_word():
2     words = input("Enter list of words separated by spaces: ").split()
3     longest_length = 0
4     for word in words:
5         if len(word) > longest_length:
6             longest_length = len(word)
7     return longest_length
8
9 result = find_longest_word()
10 print(result)
```

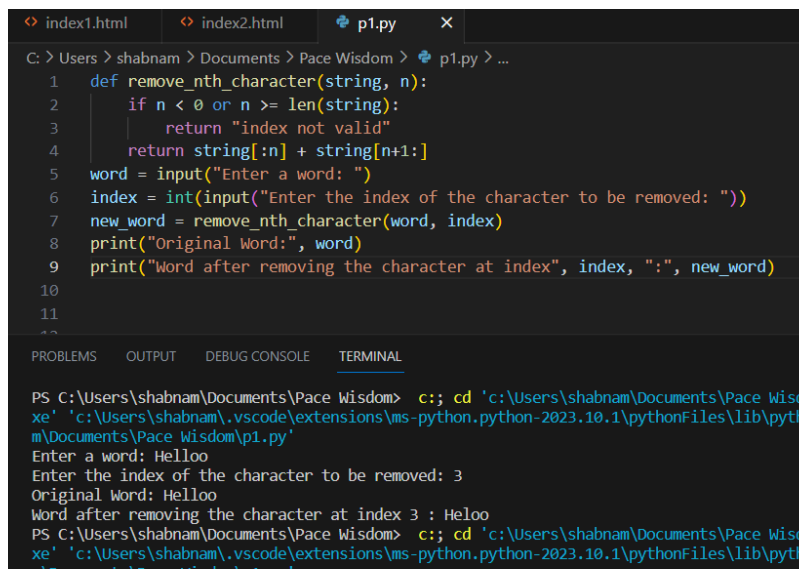
Below the editor, the TERMINAL panel shows the command prompt output:

```
PS C:\Users\shabnam\Documents\Pace Wisdom> c::; cd 'c:\Users\shabnam\Documents\Pace Wisdom'
Enter list of words separated by spaces: hellooooooooooooo good morning
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)
```



The screenshot shows a VS Code editor with a file named `p1.py` open. The code in the editor is as follows:

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

Below the editor, the `TERMINAL` tab is active, showing the command prompt output:

```
PS C:\Users\shabnam\Documents\Pace Wisdom> c::; cd 'c:\Users\shabnam\Documents\Pace Wisdom'
Enter a word: Hello
Enter the index of the character to be removed: 3
Original Word: Hello
Word after removing the character at index 3 : Heloo
PS C:\Users\shabnam\Documents\Pace Wisdom> c::; cd 'c:\Users\shabnam\Documents\Pace Wisdom'
```

10. Write a Python program that accepts a comma separated sequence of words as input and prints the unique words in sorted form (alphanumerically).

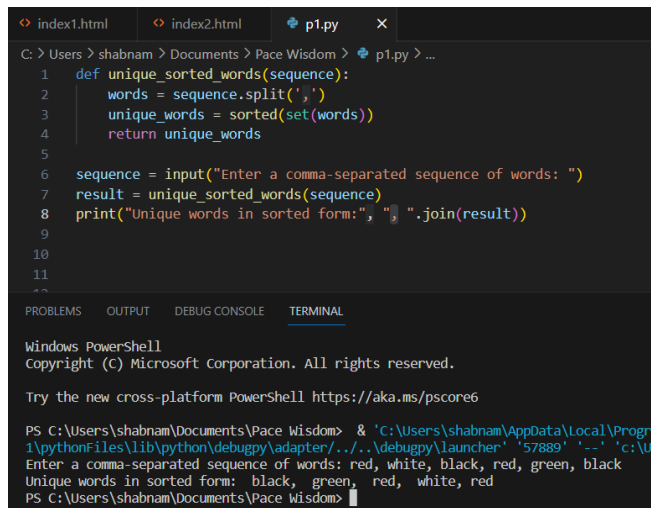
Sample Words : red, white, black, red, green, black

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))
```



The screenshot shows a Python IDE with a file named 'p1.py' open. The code in the editor is as follows:

```
1 def unique_sorted_words(sequence):  
2     words = sequence.split(',')  
3     unique_words = sorted(set(words))  
4     return unique_words  
5  
6 sequence = input("Enter a comma-separated sequence of words: ")  
7 result = unique_sorted_words(sequence)  
8 print("Unique words in sorted form:", " , ".join(result))  
9  
10  
11  
12
```

The IDE has tabs for 'index1.html', 'index2.html', and 'p1.py'. Below the editor, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'TERMINAL' tab is active, showing the following output:

```
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\Programs\Python\Python38-64\python.exe' -i -c 'import sys; sys.argv[1:];' '57889' '--' 'c:\Users\shabnam\Documents\Pace Wisdom\p1.py'  
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>
```

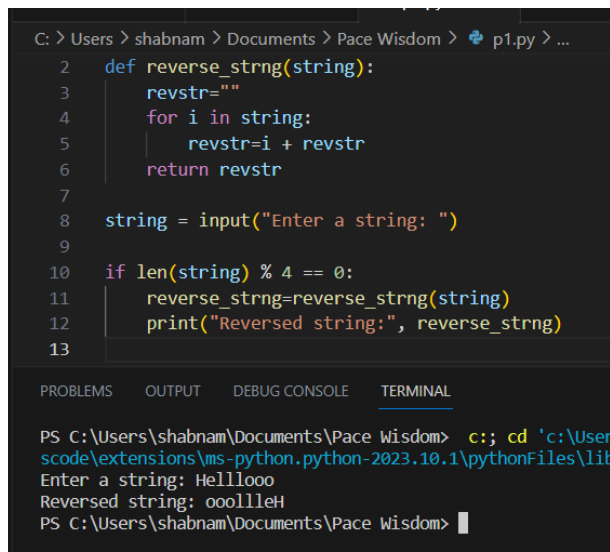
11. Write a Python function to reverse a string if its length is a multiple of 4.

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)
```



The screenshot shows a Python IDE with a dark theme. The editor window displays the code for reversing a string. The file path at the top is 'C:\Users\shabnam\Documents\Pace Wisdom > p1.py > ...'. The code is as follows:

```
2 def reverse_strng(string):  
3     revstr=""  
4     for i in string:  
5         revstr=i + revstr  
6     return revstr  
7  
8 string = input("Enter a string: ")  
9  
10 if len(string) % 4 == 0:  
11     reverse_strng=reverse_strng(string)  
12     print("Reversed string:", reverse_strng)  
13
```

Below the editor, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'TERMINAL' tab is active, showing the command prompt output:

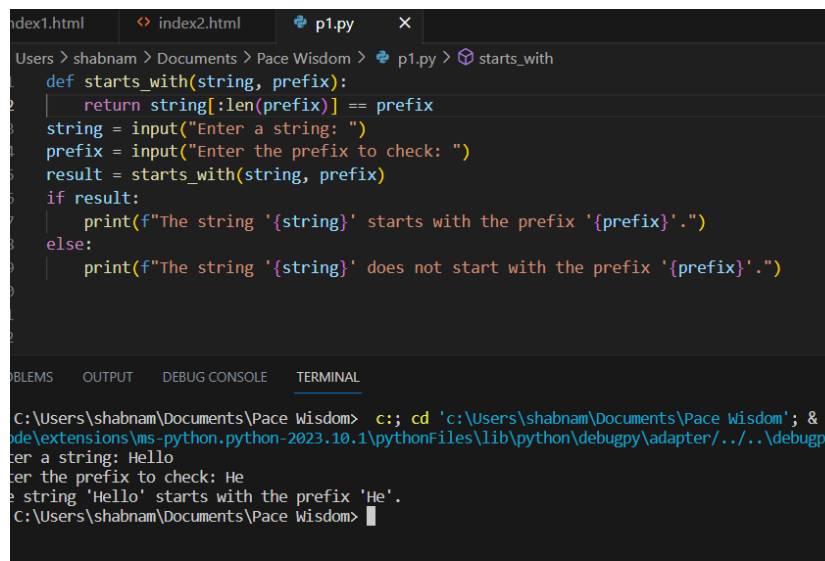
```
PS C:\Users\shabnam\Documents\Pace Wisdom> c:: cd 'c:\User  
scode\extensions\ms-python.python-2023.10.1\pythonFiles\lib  
Enter a string: Helllooo  
Reversed string: ooolllleH  
PS C:\Users\shabnam\Documents\Pace Wisdom>
```

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}'.")
```



The screenshot shows a Python IDE with a file named p1.py. The code in the file is the same as the one provided in the previous block. The terminal output shows the following sequence of events: the user enters 'Hello' for the string and 'He' for the prefix, and the program outputs 'The string 'Hello' starts with the prefix 'He'.'

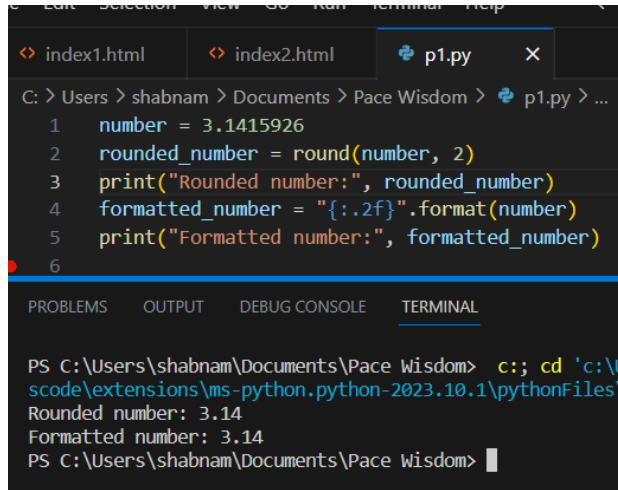
14. Write a Python program to print the following floating numbers upto 2 decimal places.
3.1415926

ANS:

```
number = 3.1415926
```



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

A screenshot of the Visual Studio Code editor interface. The top panel shows three tabs: 'index1.html', 'index2.html', and 'p1.py'. The 'p1.py' tab is active, displaying a Python script with six lines of code. The bottom panel is divided into 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL' tabs, with 'TERMINAL' being the active tab. The terminal shows the command prompt 'PS C:\Users\shabnam\Documents\Peace Wisdom>' followed by the execution of 'c:; cd 'c:\Users\shabnam\Documents\Peace Wisdom>' and the output of the Python script: 'Rounded number: 3.14' and 'Formatted number: 3.14'.

```
C: > Users > shabnam > Documents > Peace Wisdom > p1.py > ...
1  number = 3.1415926
2  rounded_number = round(number, 2)
3  print("Rounded number:", rounded_number)
4  formatted_number = "{:.2f}".format(number)
5  print("Formatted number:", formatted_number)
6

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\shabnam\Documents\Peace Wisdom> c:; cd 'c:\Users\shabnam\Documents\Peace Wisdom'
Rounded number: 3.14
Formatted number: 3.14
PS C:\Users\shabnam\Documents\Peace 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)
```

```
index1.html  index2.html  p1.py  X

C: > Users > shabnam > Documents > Pace Wisdom > p1.py > ...
1  def repeated_character(string):
2      char_count = {}
3      for char in string:
4          if char in char_count:
5              char_count[char] += 1
6          else:
7              char_count[char] = 1
8      for char, count in char_count.items():
9          if count > 1:
10             print(char, count)
11 sample_string = 'thequickbrownfoxjumpsoverthelazydog'
12 repeated_character(sample_string)
13

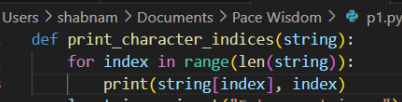
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\shabnam\Documents\Pace Wisdom> c++ cd 'c:\Users\shabnam\Documents\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)
```



```
C:\Users\shabnam\Documents\Peace Wisdom> c:\> cd 'c:\Users\shabnam\Documents\Peace Wisdom> python p1.py
```

```
def print_character_indices(string):
    for index in range(len(string)):
        print(string[index], index)

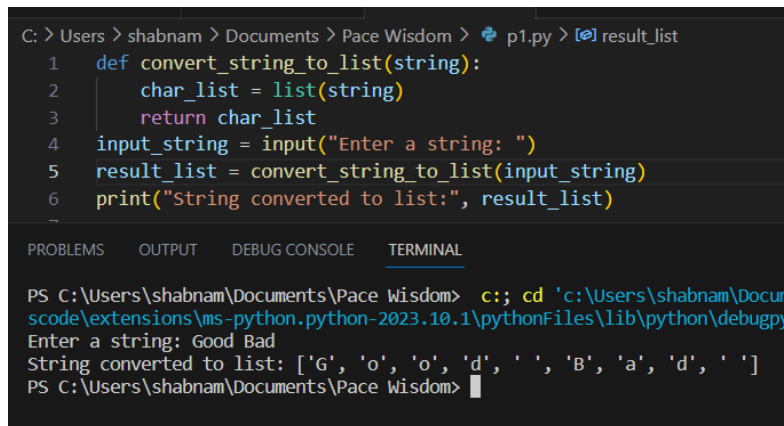
sample_string = input("Enter a string: ")
print_character_indices(sample_string)
```

```
PS C:\Users\shabnam\Documents\Peace Wisdom> c:\> cd 'c:\Users\shabnam\Documents\Peace Wisdom> python p1.py
Enter a string: Hello
H 0
e 1
l 2
l 3
o 4
PS C:\Users\shabnam\Documents\Peace 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)
```



The screenshot shows a VS Code editor window with a Python file named p1.py. The code defines a function convert_string_to_list that takes a string and returns a list of its characters. It then prompts the user to enter a string, converts it to a list, and prints the result. The terminal output shows the user entering 'Good Bad' and the program printing the list ['G', 'o', 'o', 'd', ' ', 'B', 'a', 'd', ' '].

```
C: > Users > shabnam > Documents > Pace Wisdom > p1.py > result_list  
1 def convert_string_to_list(string):  
2     char_list = list(string)  
3     return char_list  
4 input_string = input("Enter a string: ")  
5 result_list = convert_string_to_list(input_string)  
6 print("String converted to list:", result_list)  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  
PS C:\Users\shabnam\Documents\Pace Wisdom> c:: cd 'c:\Users\shabnam\Documents\Pace Wisdom' & python p1.py  
Enter a string: Good Bad  
String converted to list: ['G', 'o', 'o', 'd', ' ', 'B', 'a', 'd', ' ']  
PS C:\Users\shabnam\Documents\Pace Wisdom>
```

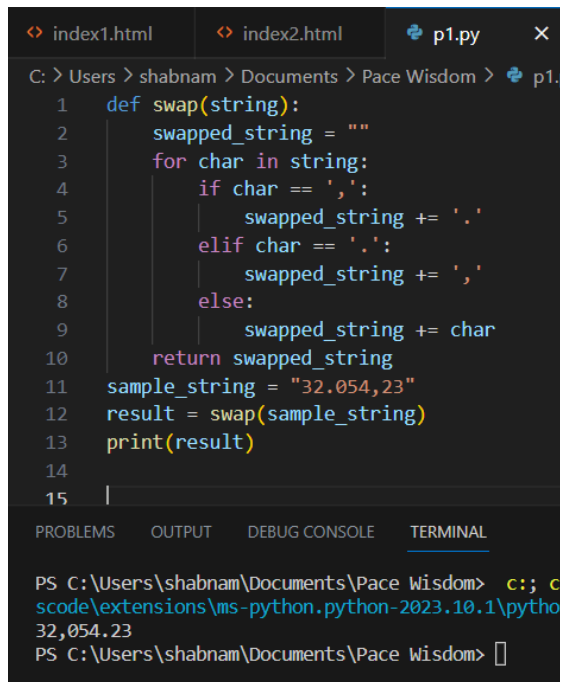
18. Write a Python program to swap comma and dot in a string.

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)
```



The screenshot shows a code editor with three tabs: index1.html, index2.html, and p1.py. The active tab is p1.py, which contains the following Python code:

```
1 def swap(string):
2     swapped_string = ""
3     for char in string:
4         if char == ',':
5             swapped_string += '.'
6         elif char == '.':
7             swapped_string += ','
8         else:
9             swapped_string += char
10    return swapped_string
11 sample_string = "32.054,23"
12 result = swap(sample_string)
13 print(result)
14
15
```

Below the code editor, the terminal output is visible:

```
PS C:\Users\shabnam\Documents\Pace Wisdom> c:: c
code\extensions\ms-python.python-2023.10.1\python
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:

```
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)
```

```
index1.num | index2.num | p1.py x
C: > Users > shabnam > Documents > Pace Wisdom > p1.py > ...
1  def smallest_and_largest_word(string):
2      words = string.split()
3      largest_word = max(words, key=len)
4      smallest_word = min(words, key=len)
5      return largest_word, smallest_word
6
7  string = input("Enter a string: ")
8  smallest, largest = smallest_and_largest_word(string)
9  print("Smallest word:", smallest)
10 print("Largest word:", largest)
11

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\shabnam\Documents\Pace Wisdom> c;; cd 'c:\Users\shabnam\Documents\Pace Wisdom>
scode\extensions\ms-python.python-2023.10.1\pythonFiles\lib\python\
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:

```
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 x
C: > Users > shabnam > Documents > Pace Wisdom > p1.py > remove_co
1 def remove_consecutive_duplicates(string):
2     result = ""
3     previous_char = None
4     for char in string:
5         if char != previous_char:
6             result += char
7             previous_char = char
8     return result
9
10 sample_string = input("Enter a string: ")
11 print(remove_consecutive_duplicates(sample_string))
12

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
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> 
```