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

```
string = "Hello, world!"

length = len(string)

print("The length of the string is:", 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}

```
 string = "google.com"

freq = {}

for char in string:

  if char in freq:

     freq[char] += 1

  else:

     freq[char] = 1

print("Character frequency in the string:", freq)
```

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

```
string = "thisisniceone"

f len(string) < 2:

   new_string = ""

else:

   new_string = string[0:2] + string[-2:]

print("New string:", new_string)

New string: then

string = "ab"

string = "f"
```

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'

```
string = "restart"

first_char = string[0]

new_string = first_char

for char in string[1:]:

   if char == first_char:

      new_string += "$"

   else:

      new_string += char

print("New string:", new_string)
```

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'

```
string1 = "abc"

string2 = "xyz"

new_string1 = string2[:2] + string1[2:]

new_string2 = string1[:2] + string2[2:]

new_string = new_string1 + " " + new_string2

print("New string:", new_string)
```

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'

```
string = "abc"
if len(string) < 3:
   new_string = string
elif string[-3:] == "ing":
   new_string = string + "ly"
else:
   new_string = string + "ing"
print("New string:", new_string)
```

EXPECTED OUTPUT:abcing

or

```
string = "string"
if len(string) < 3:
   new_string = string
elif string[-3:] == "ing":
   new_string = string + "ly"
else:
   new_string = string + "ing"
print("New string:", new_string)
```

EXPECTED OUTPUT:stringly

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!'

```
string = "The lyrics is not that poor!"
index_not = string.find("not")
index_poor = string.find("poor")
if index_poor > index_not and index_not != -1 and index_poor != -1:
    new_string = string[:index_not] + "good" + string[index_poor+4:]
else:
    new_string = string
print("New string:", new_string)
```

EXPECTED OUTPUT:

'The lyrics is good!'

or

```
string = "The lyrics is poor!"
index_not = string.find("not")
index_poor = string.find("poor")
if index_poor > index_not and index_not != -1 and index_poor != -1:
    new_string = string[:index_not] + "good" + string[index_poor+4:]
else:
    new_string = string
print("New string:", new_string)
```

EXPECTED OUTPUT:

New string: The lyrics is poor!

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

```
def longest_word(words):
   max_len = 0
   for word in words:
      if len(word) > max_len:
         max_len = len(word)
   return max_len
```

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

```
string = "Hello, world!"
n = 5
new_string = remove_char(string, n)
print(new_string)
```

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

```
def longest_word(words):
   max_len = 0
   for word in words:
      if len(word) > max_len:
         max_len = len(word)
   return max_len
words = ['apple', 'banana', 'cherry', 'durian', 'elderberry']
print(longest_word(words))
```

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

```
def reverse_string_if_multiple_of_4(string):
    if len(string) % 4 == 0:
        return string[::-1]
    else:
        return string
string1 = 'abcd'
string2 = 'abcdef'
string3 = 'python'
print(reverse_string_if_multiple_of_4(string1))  # Output: 'dcba'
print(reverse_string_if_multiple_of_4(string2))  # Output: 'abcdef'
print(reverse_string_if_multiple_of_4(string3))  # Output: 'python'
```

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.

```
def convert_to_uppercase(string):
    uppercase_count = 0
    for char in string[:4]:
        if char.isupper():
            uppercase_count += 1
    if uppercase_count >= 2:
        return string.upper()
    else:
        return string
string1 = 'HELLO world!'
string2 = 'Hello World!'
string3 = 'HeLLo world!'
```

```
print(convert_to_uppercase(string1))  # Output: 'HELLO WORLD!'

print(convert_to_uppercase(string2))  # Output: 'Hello World!'

print(convert_to_uppercase(string3))  # Output: 'HeLLo world!'
```

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

```
def starts_with(string, start):
   if string.startswith(start):
      return True
   else:
      return False
string1 = 'Hello World!'

string2 = 'Goodbye World!'

print(starts_with(string1, 'He'))  # Output: True

print(starts_with(string1, 'Hi'))  # Output: False

print(starts_with(string2, 'Go'))  # Output: True

print(starts_with(string2, 'No'))  # Output: False
```

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

```
num = 3.1415926

print("{:.2f}".format(num))
```

3.14

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

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

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

```
string = 'Hello, world!'
for i in range(len(string)):
    print(string[i], i)
```

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

```
string = 'hello world'

string_list = list(string)

print(string_list)
```

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

Sample string: "32.054,23"

Expected Output: "32,054.23"

```
string = "32.054,23"

table = str.maketrans({",": ".", ".": ","})

new_string = string.translate(table)

print(new_string)
```

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

```
string = "The quick brown fox jumps over the lazy dog"

words = string.split()

smallest_word = words[0]

largest_word = words[0]

for word in words:

   if len(word) < len(smallest_word):

      smallest_word = word

   if len(word) > len(largest_word):

      largest_word = word

print("Smallest word:", smallest_word)

print("Largest word:", largest_word)
```

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

```
def remove_consecutive_duplicates(s):
    """
    Remove all consecutive duplicates of a given string.
    Parameters:
    s (str): The input string.
    Returns:
    str: The modified string with all consecutive duplicates removed.
    """
    if not s:
        return ''
    result = [s[0]]
    for i in range(1, len(s)):
        if s[i] != s[i-1]:
            result.append(s[i])
    return ''.join(result)
s = 'aabbccdefgghhii'
result = remove_consecutive_duplicates(s)
print(result)  # 'abcdefghi'
```