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

```
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}

```
def nos_of_char(str):
    dict = {}
    for n in str1:
        keys = dict.keys()
        if n in keys:
            dict[n] += 1
        else:
            dict[n] = 1
    return dict
print(nos_of_char('google.com'))
```

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

```
def both_ends(str):
    if len(str) < 2:
        return ''
    return str[0:2] + str[-2:]
print(both_ends('thisisniceone'))
```

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'

```
str1 = 'restart'
char = str1[0]
str1 = str1.replace(char, '$')
str1 = char + str1[1:]
print(str1)
```

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'

```
a = 'abc'
b = 'xyz'
new_a = b[:2] + a[2:]
new_b = a[:2] + b[2:]
print(new_a + ' ' + new_b)
```

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'

```
def add_string(str1):
   length = len(str1)
   if length > 2:
      if str1[-3:] == 'ing':
         str1 += 'ly'
      else:
         str1 += 'ing'
   return str1
print(add_string('abc'))
print(add_string('string'))
```

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!'

```
def not_poor(str1):
   snot = str1.find('not')
   spoor = str1.find('poor')
   if spoor > snot and snot > 0 and spoor > 0:
      str1 = str1.replace(str1[snot:(spoor+4)], 'good')
   return str1

print(not_poor('The lyrics is not that poor!'))
print(not_poor('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_list):
   word_len = []
   for n in words_list:
      word_len.append((len(n), n))
   word_len.sort()
```

```
    return word_len[-1][1]

print(longest_word(["PHP", "Exercises", "Backend"]))
```

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

```
def remove_char(str, n):
    first_part = str[:n]
    last_part = str[n+1:]
    return first_part + last_part

print(remove_char("Python", 0))
print(remove_char("Python", 3))
print(remove_char("Python", 5))
```

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

```
items = input("Input comma separated sequence of words")
words = [word for word in items.split(",")]
print(",".join(sorted(list(set(words)))))
```

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

```
def reverse_string(str1):
    if len(str1) % 4 == 0:
        return ''.join(reversed(str1))
    return str1
print(reverse_string('abcd'))
print(reverse_string('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 to_uppercase(str1):
    num_upper = 0
    for letter in str1[:4]:
        if letter.upper() == letter:
            num_upper += 1
    if num_upper >= 2:
        return str1.upper()
    return str1
print(to_uppercase('Python'))
print(to_uppercase('PyThon'))
```

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

```
def start_specified_chars(str1):
    print(str1.startswith("Pa"))
start_specified_chars("Shreesha")
```

```
start_specified_chars("Pacewisdom")
```

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

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

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

```
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)
```

OR

```
import collections
str1 = 'thequickbrownfoxjumpsoverthelazydog'
d = collections.defaultdict(int)
for c in str1:
    d[c] += 1

for c in sorted(d, key=d.get, reverse=True):
  if d[c] > 1:
      print('%s %d' % (c, d[c]))
```

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

```
str1 = "Python"
for index, char in enumerate(str1):
    print("Current character", char, "position at", index)
```

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

```
print(list("Python"))
```

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

```
def swap_comma_dot(input_string):
    swapped_string = input_string.translate(str.maketrans(',.', '.,'))
    return swapped_string

sample_string = "32.054,23"
swapped_string = swap_comma_dot(sample_string)
print(swapped_string)
```

```
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)
```

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

```
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)
```

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

```
from itertools import groupby
def remove_all_consecutive(str1):
    result_str = []
    for (key,group) in groupby(str1):
        result_str.append(key)
```

```
    return ''.join(result_str)

print(remove_all_consecutive("pythonnnnn"))
print(remove_all_consecutive("shreeshasa"))
```

OR

```
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))
```