1.To calculate length of a string

```
def calculate_length(input):
     length = len(input)
     return length
 input = input("Enter a string: ")
 string_length = calculate_length(input)
print("The length of the string is: {string_length}")
```

2.Count the number of characters in a string

```
def count_character(input_string):
    char_freq   = {}
     for char in input_string:
        if char in char_freq:
            char_freq[char] += 1
        else:
            char_freq[char] = 1
    return char_freq
sample_string = "google.com"
result = count_character(sample_string)
print(result)
```

3. Get string with first 2 and last 2 chars

```
def get_modified_string(input_string):
    if len(input_string) < 2:
        return " "
    else:
        return input_string[:2] + input_string[-2:]
sample_string = "thisisniceone"
modified_string = get_modified_string(sample_string)
```

```
print(modified_string)
```

4. Replace occurrences of first char in a string

```
def replace_first_char_occurrences(input_string):
    first_char = input_string[0]
    modified_string = first_char + input_string[1:].replace(first_char, "$")
    return modified_string
sample_string = "restart"
modified_string = replace_first_char_occurrences(sample_string)
print(modified_string)
```

5. Swap first two characters of two strings

```
 def swap(string1, string2):
     new_string1 = string2[:2] + string1[2:]
     new_string2 = string1[:2] + string2[2:]
     return new_string1 + " " + new_string2
 string1 = "abc"
 string2 = "xyz"
 swapped_string = swap(string1, string2)
 print(swapped_string)
```

6. Add 'ing' or 'ly' to a string

```
def modify_string(input_string):
    if len(input_string) < 3:
        return input_string
    elif input_string[-3:] == "ing":
        return input_string + "ly"
    else:
        return input_string + "ing"
```

```
sample_string = input("enter an input string")
modified_string = modify_string(sample_string)
print(modified_string)
```

7. Replace 'not'...'poor' with 'good'

```
def replace(input_string):
    not = input_string.find("not")
    poor = input_string.find("poor")
    if not != -1 and poor != -1 and poor> not:
        return input_string[:not] + "good" + input_string[poor+4:]
    else:
        return input_string
sample_string = input(“enter the input string”)
modified_string = replace(sample_string)
print(modified_string)
```

8. Find longest word in a list

```
def find_longest_word(words):
    longest = max(words, key=len)
    return longest
words= ["apple", "banana", "cherry", "date"]
longest_word = find_longest_word(words)
print("Longest word:", longest_word)
```

9. Remove nth index character from string

```
def remove_nth_char(input_string, n):
    if n < 0 or n >= len(input_string):
        return input_string
    return input_string[:n] + input_string[n+1:]
```

```
sample_string = "programming"
n = 4
modified_string = remove_nth_char(sample_string, n)
print(modified_string)
```

10. Print unique words in sorted form

```
def unique_sorted_words(input_string):
    words = input_string.split(", ")
    unique_words = sorted(set(words))
    return unique_words
sample_words = "red, white, black, red, green, black"
result = unique_sorted_words(sample_words)
print(result)
```

11. Reverse string if its length is multiple of 4

```
def reverse(input_string):
    if len(input_string) % 4 == 0:
        return input_string[::-1]
    return input_string
sample_string = "abcdefgh"
reversed_string = reverse(sample_string)
print(reversed_string)
```

12. Convert string to uppercase if contains 2 uppercase in first 4 chars

```
def convert(input_string):
    uppercase_count = sum(1 for char in input_string[:4] if char.isupper())
    if uppercase_count >= 2:
        return input_string.upper()
    return input_string
```

```
sample_string = input("enter an input string")
converted_string = convert(sample_string)
print(converted_string)
```

13. Check if string starts with specified characters

```
def starts_with(input_string, specified_chars):
    return input_string.startswith(specified_chars)
sample_string = "Python is great"
specified_chars = "Py"
starts_with_result = starts_with(sample_string, specified_chars)
print(starts_with_result)
```

14. Print floating numbers with 2 decimal places

```
floating_number = 3.1415926
print("{:.2f}".format(floating_number))
```

15. Count repeated characters in a string

```
def count_repeated(input_string):
    char_frequency = {}
    for char in input_string:
        if char in char_frequency:
            char_frequency[char] += 1
        else:
            char_frequency[char] = 1
 repeated_chars = {char: freq for char, freq in char_frequency.items() if freq > 1}
    return repeated_chars
sample_string = ("enter the sample string")
repeated_chars_result = count_repeated(sample_string)
for char, freq in repeated_chars_result.items():
    print(char, freq)
```

16. Print index of characters in a string

```
def indices(input_string):
    for index, char in enumerate(input_string):
        print("Character '{char}' found at index {index}")
sample_string = input(“enter a string")
indices(sample_string)
```

17. Convert string to list of characters

```
def convert(input_string):
    return list(input_string)
sample_string = ("enter a string")
char_list = convert(sample_string)
print(char_list)
```

18. Swap comma and dot in a string

```
def swap(input_string):
    return input_string.translate(str.maketrans(",.", ".,"))
sample_string = "32.054,23"
swapped_string = swap(sample_string)
print(swapped_string)
```

19. Find smallest and largest word in a string

```
def find(input_string):
    words_list = input_string.split()
    smallest_word = min(words_list, key=len)
    largest_word = max(words_list, key=len)
    return smallest_word, largest_word
```

```
sample_string = input("enter a string")
smallest_word, largest_word = find(sample_string)
print("Smallest word:", smallest_word)
print("Largest word:", largest_word)
```

20. Remove consecutive duplicates from a string

```
def remove(input_string):
    result = []
    prev_char = None
    for char in input_string:
        if char != prev_char:
            result.append(char)
            prev_char = char
    return ''.join(result)
sample_string = input("enter a string")
modified_string = remove(sample_string)
printf(modified_string)
```