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

```
    str=input("Enter the string :")

    print(len(str));
```

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 char_frequency(str1):

    dict = {}

    for n in str1:

        keys = dict.keys()

        if n in keys:

            dict[n] += 1

        else:

            dict[n] = 1

    return dict

str=input("Enter the string :")

print(char_frequency(str))
```

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”'

```
def string_both_ends(str):

  if len(str) < 2:

    return ''
  return str[0:2] + str[-2:]
str=input("Enter the string :")
print(string_both_ends(str))
```

Sample String : 'ab'

Expected Result : 'abab'

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

Sample String : 'f'

Expected Result : Empty String

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

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

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'
   print("Before Swap :",a," ",b)
   a1 = b[:2] + a[2:]
   b1 = a[:2] + b[2:]
   print("After Swap :",a1," ",b1)
```

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'

```
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'))
```

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('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
  else:
    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 find_longest_word(words_list):
    word_len = []
    for n in words_list:
        word_len.append((len(n), n))
    word_len.sort()
    return word_len[-1][0], word_len[-1][1]
result = find_longest_word(["Python","Programming", "Language",])
print("\nLongest word: ", result[1])
print("Length of the longest word: ", result[0])
```

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.

```
str=input("enterthe word:")
if(len(str)%4==0):
   print(str[::-1])
else:
   print("cant")
```

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.

```
str1 = input("Enter the String :")
count = 0
for letter in str1[:4]:
   if letter.upper() == letter:
      count += 1
if count >= 2:
    print(str1.upper())
print(str1)
```

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

```
string1=input("Enter the string : ")
string2=input("Enter the character :")
if(string1.startswith(string2)):
    print("Yes It starts with "+string2)
else:
    print("No it does not starts with "+string2)
```

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

3.1415926

```
num=3.1415926
print (num)
print(format(num,".2f"))
```

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

```
import collections
string1 = input("enter the string")
d=collections.defaultdict(int)
for x in string1:
    d[x] += 1
for i in sorted(d,key=d.get,reverse = True):
    if  d[i] >1:
        print('%s %d' % (i,d[i]))
```

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

```
stringp = input("Enter the string :")
stringc = input("Enter the character :")
print(stringp.index(stringc))
```

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

```
s = input("Enter the string")
print(f'List of Words ={s.split()}')
```

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

Sample string: "32.054,23"

Expected Output: "32,054.23"

```
a = input("Enter the string")
print(a)
maketrans = a.maketrans
```

```
a = a.translate(maketrans(',.', '.,'))

print(a)
```

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

```
sentence = input("Enter the String: ")

print("\nLargest Word(s): ")

word=[len(k) for k in sentence.split()]

print(*[i for i in sentence.split() if len(i) == max(word)])

print("\nSmallest Word(s): ")

print(*[i for i in sentence.split() if len(i) == min(word)])
```

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

```
class Solution:

  def solve(self, s):

      seen = s[0]

      ans = s[0]

      for i in s[1:]:

         if i != seen:

            ans += i

            seen = i

      return ans

ob = Solution()

str=input("Enter the string : ")

print(ob.solve(str))
```