

PYTHON STRING ASSIGNMENT

(MANEESH S SHETTY)

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

```
s=input('enter the string :')
count=0
for i in s:
    count=count+1
print(count)
enter the string :maneesh
7
```

2. Write a Python program to count the number of characters (character frequency) in a string.

```
a = input("Enter the string: ")
di = {}
for i in a:
    if i in di:
        di[i] += 1
    else:
        di[i] = 1
print(di)
Enter the string: strings
{'s': 2, 't': 1, 'r': 1, 'i': 1, 'n': 1, 'g': 1}
```

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

```
s = input("enter the string")
if (len(s)>1):
    a=list(s)
    b=s[0:2]
    c=s[-1:-3:-1]
    d=c[-1:-3:-1]
    e=b+d
    print(e)
```

```
else:  
    print("string should be greater than 1")
```

```
enter the stringmaneesh shetty  
matty
```

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'

```
i=input("enter the string")  
a=i[0]  
b=i[1:]  
lst=[]  
lst.append(a)  
for i in b:  
    if i==a:  
        lst.append("$")  
    else:  
        lst.append(i)  
c=" ".join(lst)  
print(c)
```

```
enter the stringrestart  
r e s t a $ t
```

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'

```
s1 = input("Enter the first string: ")  
s2 = input("Enter the second string: ")  
result = s2[:2] + s1[2:] + " " + s1[:2] + s2[2:]  
print(result)
```

```
Enter the first string: python  
Enter the second string: language  
lathon pynguage
```

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'

```
a=input("enter the string")
if (len(a)>2):
    if a[-1:-4:-1]== "gni":
        print(a+"ly")
    else:
        print(a+"ing")
enter the stringstring
stringly
enter the stringfloat
floating
```

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 replace_not_poor(s):
    not_pos = s.find('not')
    poor_pos = s.find('poor')
    if not_pos != -1 and poor_pos != -1 and not_pos < poor_pos:
        result = s[:not_pos] + 'good' + s[poor_pos + 4:]
    return result
return s
```

s1 = 'The lyrics is not that poor!'

print(f"Input: '{s1}'")

print(f"Output: '{replace_not_poor(s1)}'")

print()

s2 = 'The lyrics is poor!'

print(f"Input: '{s2}'")

print(f"Output: '{replace_not_poor(s2)}'")

```
Input: 'The lyrics is not that poor!
Output: 'The lyrics is good! '
```

```
Input: 'The lyrics is poor!'
Output: '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_length(words):
```

```
    if not words:
```

```
        return 0
```

```
return max(len(word) for word in words)

def longest_word_length_v2(words):
    if not words:
        return 0

    max_length = 0
    for word in words:
        if len(word) > max_length:
            max_length = len(word)
    return max_length

words1 = ['python', 'is', 'awesome']
print(f'Words: {words1}')
print(f'Longest word length: {longest_word_length(words1)}')
print()

words2 = ['a', 'bb', 'ccc', 'dddd']
print(f'Words: {words2}')
print(f'Longest word length: {longest_word_length(words2)}')
print()

words3 = ['programming']
print(f'Words: {words3}')
print(f'Longest word length: {longest_word_length(words3)}')
print()

words4 = []
print(f'Words: {words4}')
print(f'Longest word length: {longest_word_length(words4)}')
print()

print("== Method 2: Using loop ==")
words5 = ['hello', 'world', 'python', 'code']
print(f'Words: {words5}')
print(f'Longest word length: {longest_word_length_v2(words5)})
```

```
Words: ['python', 'is', 'awesome']
Longest word length: 7

Words: ['a', 'bb', 'ccc', 'dddd']
Longest word length: 4

Words: ['programming']
Longest word length: 11

Words: []
Longest word length: 0

== Method 2: Using loop ==
Words: ['hello', 'world', 'python', 'code']
Longest word length: 6
```

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

```
def remove1(str1,n):
    str2 =[]
    for i in range(len(str1)):
        if i+1 ==n:
            pass
        else :
            str2.append(str1[i])
    return " ".join(str2)
str1=input("enter the string")
n=int(input("enter the position "))
result=remove1(str1,n)
print(result)
● enter the stringpython
    enter the position 2
    p t h o n
```

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)))))
```

```
Q11.py
```

```
Input comma-separated sequence of wordsred,green,black,red,white  
black,green,red,white
```

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

```
s=input("Enter the String :")  
if len(s) % 4 == 0:  
    print(".join(reversed(s)))  
else:  
    print(s)
```

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'))
```

```
Python
PYTHON
```

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

```
string1 = input("enter the string")
print(string1.startswith("py"))
```

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

```
3.1415926
x = 3.1415926
y = 12.9999
print()
print("Original Number: ", x)
print("Formatted Number: "+"{:.2f}".format(x))
print("Original Number: ", y)
print("Formatted Number: "+"{:.2f}".format(y))
print()
```

```
Original Number: 3.1415926
Formatted Number: 3.14
Original Number: 12.9999
Formatted Number: 13.00
```

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
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]))
```

```
o 4
e 3
t 2
h 2
u 2
r 2
```

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

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

```
pace.py
current character p position at 0
current character a position at 1
current character c position at 2
current character e position at 3
current character position at 4
Current character w position at 5
Current character i position at 6
Current character s position at 7
Current character d position at 8
Current character o position at 9
Current character m position at 10
```

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

```
str1=input("enter a string")
print(list(str1))
enter a stringhello
['h', 'e', 'l', 'l', 'o']
```

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

Sample string: "32.054,23"

```
Expected Output: "32,054.23"
amount = "32.054,23"
maketrans = amount.maketrans
amount = amount.translate(maketrans(., ','))
print(amount)
amount1 = "32.054,23"
```

```
amount1 = amount1.replace('.', '#')
amount1 = amount1.replace(',', '.')
amount1 = amount1.replace('#', ',')
print(amount1)
```

```
32,054.23
32,054.23
```

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

```
str1="a man is alive"
```

```
min=100
```

```
max=-1
```

```
letter_min=" "
```

```
letter_max=""
```

```
for i in str1.split(" "):
```

```
 min1=len(i)
```

```
 if min1<min:
```

```
 min=min1
```

```
 letter_min=i
```

```
for i in str1.split(" "):
```

```
 max1=len(i)
```

```
 if max1>max:
```

```
 min=min1
```

```
 letter_max=i
```

```
print(letter_min)
```

```
print(letter_max)
```

```
a
alive
```

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

```
import itertools
```

```
def remove_consecutive_duplicates(s1):
 return ''.join(i for i, _ in itertools.groupby(s1))
s1= "aabdaee"
print("Original String: ",s1)
print("\n Removing all consecutive duplicates:")
print(remove_consecutive_duplicates(s1))
```

0123.py

Original string: aabcdcae

Removing all consecutive duplicates:

abcdae