

## **06 - Strings in Python**





**Ex. No. : 6.1**

**Date: 04.05.2024**

**Register No.: 230701258**

**Name: Ram Pratheesh S.K**

## **Count Chars**

Write a python program to count all letters, digits, and special symbols respectively from a given string

For example:

| Input | Result |
|-------|--------|
|-------|--------|

|         |  |
|---------|--|
| rec@123 |  |
|---------|--|

|   |  |
|---|--|
| 3 |  |
|---|--|

|   |  |
|---|--|
| 3 |  |
|---|--|

|   |  |
|---|--|
| 1 |  |
|---|--|

```
a=input()
c,d,s=0,0,0
for i in range(len(a)):
    if(a[i].isalpha()):
        c+=1
    elif(a[i].isdigit()):
        d+=1
    else:
        s+=1
print(c,d,s,sep="\n")
```



|   | Input           | Expected    | Got         |   |
|---|-----------------|-------------|-------------|---|
| ✓ | rec@123         | 3<br>3<br>1 | 3<br>3<br>1 | ✓ |
| ✓ | P@#yn26at^&i5ve | 8<br>3<br>4 | 8<br>3<br>4 | ✓ |
| ✓ | abc@12&         | 3<br>2<br>2 | 3<br>2<br>2 | ✓ |

Passed all tests! ✓



Ex. No. : 6.2

Date: 04.05.2024

Register No.: 230701258

Name: Ram Pratheesh S.K

## Decompress the String

Assume that the given string has enough memory. Don't use any extra space(IN-PLACE)

Sample Input 1  
a2b4c6

Sample Output 1  
aabbbbcccccc

```
import re
a=input()
all=re.findall('\d+',a)
all_w=re.findall('[a-z]',a)
b=""
for i,j in zip(all,all_w):
    b+=int(i)*j
print(b)
```

|   | Input   | Expected             | Got                  |   |
|---|---------|----------------------|----------------------|---|
| ✓ | a2b4c6  | aabbbbcccccc         | aabbbbcccccc         | ✓ |
| ✓ | a12b3d4 | aaaaaaaaaaaabbbddddd | aaaaaaaaaaaabbbddddd | ✓ |

Passed all tests! ✓



**Ex. No. : 6.3**

**Date: 04.05.2024**

**Register No.: 230701258**

**Name: Ram Pratheesh S.K**

### **First N Common Chars**

Two string values S1, S2 are passed as the input. The program must print first N characters present in S1 which are also present in S2.

**Input Format:**

The first line contains S1.  
The second line contains S2.  
The third line contains N.

**Output Format:**

The first line contains the N characters present in S1 which are also present in S2.

**Boundary Conditions:**

$2 \leq N \leq 10$   
 $2 \leq \text{Length of S1, S2} \leq 1000$

**Example Input/Output 1:**

**Input:**

abcbde  
cdefghbb  
3

**Output:**

bcd

**Note:**

b occurs twice in common but must be printed only once.

```
a=input()
b=input()
C=""
d=int(input())
```



```

for i in range(len(a)):
    if(len(C)-d==0):
        break
    else:
        if(a[i]in b):
            if(a[i] not in C):
                C+=a[i]
print (C)

```

|   | Input                   | Expected | Got |   |
|---|-------------------------|----------|-----|---|
| ✓ | abcbde<br>cdefghbb<br>3 | bcd      | bcd | ✓ |

Passed all tests! ✓



Ex. No. : 6.4

Date: 04.05.2024

Register No.: 230701258

Name: Ram Pratheesh S.K

## Remove Characters

Given two Strings s1 and s2, remove all the characters from s1 which is present in s2.

Constraints

1<= string length <= 200

Sample Input 1

experience

enc

Sample Output 1

xpri

```
def remove_chars(s1, s2):  
    return ''.join([char for char in s1 if char not in s2])  
s1=input()  
s2=input()  
result = remove_chars(s1, s2)  
print(result)
```

|   | Input             | Expected | Got  |   |
|---|-------------------|----------|------|---|
| ✓ | experience<br>enc | xpri     | xpri | ✓ |

Passed all tests! ✓



Ex. No. : 6.5

Date: 04.05.2024

Register No.: 230701258

Name: Ram Pratheesh S.K

## Remove Palindrome Words

String should contain only the words are not palindrome.

Sample Input 1

Malayalam is my mother tongue

Sample Output 1

is my mother tongue

For example:

| Input                         | Expected            |
|-------------------------------|---------------------|
| Malayalam is my mother tongue | is my mother tongue |
| He did a good deed            | he good             |

```
a=[]  
a=input()  
b=a. split()  
for i in b:  
    k=i.lower()  
    if k!=k[::-1]:  
        print(k,end=' ')
```

|   | Input                         | Expected            | Got                 |   |
|---|-------------------------------|---------------------|---------------------|---|
| ✓ | Malayalam is my mother tongue | is my mother tongue | is my mother tongue | ✓ |

Passed all tests! ✓



Ex. No. : 6.6

Date: 04.05.2024

Register No.: 230701258

Name: Sailesh Rangaraj

## **Return Second Word in Uppercase**

Write a program that takes as input a string (sentence), and returns its second word in uppercase.

For example:

If input is "Wipro Technologies Bangalore" the function should return "TECHNOLOGIES"

If input is "Hello World" the function should return "WORLD"

If input is "Hello" the program should return "LESS"

NOTE 1: If input is a sentence with less than 2 words, the program should return the word "LESS".

NOTE 2: The result should have no leading or trailing spaces.

For example:

| Input                        | Result       |
|------------------------------|--------------|
| Wipro Technologies Bangalore | TECHNOLOGIES |
| Hello World                  | WORLD        |
| Hello                        | LESS         |

```
f=input()
s=f.split()
if len(s)>1:
    c=s[1]
    print(c.upper())
else:
    print("LESS")
```

|   | Input                        | Expected     | Got          |   |
|---|------------------------------|--------------|--------------|---|
| ✓ | Wipro Technologies Bangalore | TECHNOLOGIES | TECHNOLOGIES | ✓ |
| ✓ | Hello World                  | WORLD        | WORLD        | ✓ |
| ✓ | Hello                        | LESS         | LESS         | ✓ |

Passed all tests! ✓





Ex. No. : 6.7

Date: 04.05.2024

Register No.: 230701258

Name: Ram Pratheesh S.K

### Revers String

Reverse a string without affecting special characters. Given a string S, containing special characters and all the alphabets, reverse the string without affecting the positions of the special characters.

Input:

A&B

Output:

B&A

Explanation: As we ignore '&' and

As we ignore '&' and then reverse, so answer is "B&A".

For example:

Input Result

A&x#

x&A#

```
def reverse_string(s):
```

```
    s = list(s)
```

```
    l, r = 0, len(s) - 1
```

```
    while l < r:
```

```
        if not s[l].isalpha():
```

```
            l += 1
```

```
        elif not s[r].isalpha():
```

```
            r -= 1
```

```
        else:
```

```
            s[l], s[r] = s[r], s[l]
```

```
            l += 1
```

```
            r -= 1
```

```
    return "".join(s)
```

```
# Test Cases
```

```
print(reverse_string(input())) # Output: "B&A"
```

|   | Input | Expected | Got |   |
|---|-------|----------|-----|---|
| ✓ | A&B   | B&A      | B&A | ✓ |

Passed all tests! ✓



Ex. No. : 6.8

Date: 04.05.2024

Register No.: 230701258

Name: Ram Pratheesh S.K

## String characters balance Test

Write a program to check if two strings are balanced. For example, strings s1 and s2 are balanced if all the characters in the s1 are present in s2. The character's position doesn't matter. If balanced display as "true", otherwise "false".

For example:

Input Result

Yn

PYnative

True

```
a=input()
b=input()
if a in b:
    print("True")
else:
    print("False")
```

|   | Input           | Expected | Got   |   |
|---|-----------------|----------|-------|---|
| ✓ | Yn<br>PYnative  | True     | True  | ✓ |
| ✓ | Ynf<br>PYnative | False    | False | ✓ |

Passed all tests! ✓



Ex. No. : 6.9

Date: 04.05.2024

Register No.: 230701258

Name: Ram Pratheesh S.K

## Unique Names

In this exercise, you will create a program that reads words from the user until the user enters a blank line. After the user enters a blank line your program should display each word entered by the user exactly once. The words should be displayed in the same order that they were first entered. For example, if the user enters:

**Input:**

first  
second  
first  
third  
second

then your program should display:

**Output:**

first  
second  
third

```
a,c=[],[]  
for i in range(0,5):  
    b=input()  
    a.append(b)  
for i in range(len(a)):  
    if(a[i] not in c):  
        c.append(a[i])  
        print(a[i])
```

|   | Input                                       | Expected                 | Got                      |   |
|---|---|--------------------------|--------------------------|---|
| ✓ | first<br>second<br>first<br>third<br>second | first<br>second<br>third | first<br>second<br>third | ✓ |
| ✓ | rec<br>cse<br>it<br>rec<br>cse              | rec<br>cse<br>it         | rec<br>cse<br>it         | ✓ |

Passed all tests! ✓





**Ex. No. : 6.10**

**Date: 04.05.2024**

**Register No.: 230701258**

**Name: Ram Pratheesh S.K**

## **Username Domain Extension**

Given a string S which is of the format USERNAME@DOMAIN.EXTENSION, the program must print the EXTENSION, DOMAIN, USERNAME in the reverse order.

### **Input Format:**

The first line contains S.

### **Output Format:**

The first line contains EXTENSION.

The second line contains DOMAIN.

The third line contains USERNAME.

### **Boundary Condition:**

1 <= Length of S <= 100

Example Input/Output 1:

### **Input:**

vijayakumar.r@rajalakshmi.edu.in

### **Output:**

edu.in  
rajalakshmi  
vijayakumar.r

```
a = input()
ext = a.split('@')[0]
dom = a.split('@')[1].split('.')[0]
userno = a.find('.')
user = a[userno+1:]
print(user)
print(dom, end='\n')
```

```
print(ext,end='\n')
```

|   | Input          | Expected             | Got                  |   |
|---|----------------|----------------------|----------------------|---|
| ✓ | abcd@gmail.com | com<br>gmail<br>abcd | com<br>gmail<br>abcd | ✓ |

Passed all tests! ✓