06 - Strings in Python

Ex. No. : 6.1 Date:8.5.24

Register No : 231501161 Name : SUBASHINI R

,

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 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 3 1	3 3 1	~
~	P@#yn26at^&i5ve	8 3 4	8 3 4	~
~	abc@12&	3 2 2	3 2 2	~

Ex. No. : 6.2 Date:8.5.24

Register No : 231501161 Name : SUBASHINI R

,

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

Program:

```
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	aaaaaaaaaaabbbdddd	aaaaaaaaaabbbdddd	~		
Passed all tests! ✔						

Ex. No. : 6.3 Date:8.5.24

Register No : 231501161 Name : SUBASHINI R

,

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:

Example Input/Output 1:

Input:

abcbde cdefghbb

3

Output:

bcd

Note:

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

Program:

a=input()



Ex. No. : 6.4 Date:8.5.24

Register No : 231501161 Name : SUBASHINI R

,

Remove Characters

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

Program:

```
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)
```



Ex. No. : 6.5 Date:8.5.24

Register No : 231501161 Name : SUBASHINI R

,

Remove Palindrome Words

String should contain only the words are not palindrome.

Sample Input 1 Malayalam is my mother tongue

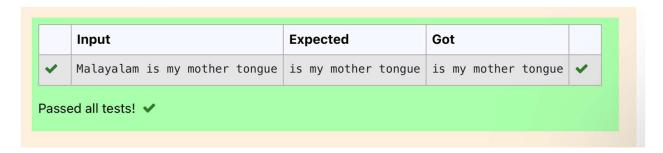
Sample Output 1 is my mother tongue

For example:

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

Program:

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



Ex. No. : 6.6 Date:8.5.24

Register No : 231501161 Name : SUBASHINI R

/

Return Second World 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"

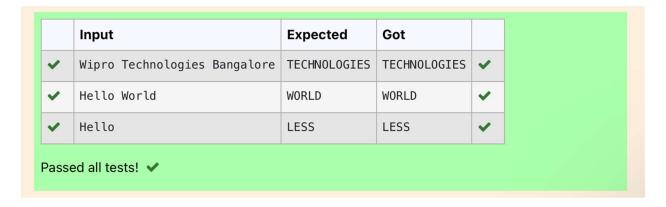
NOTE 1: If input is a sentence with less than 2 words, the program should return the word "LESS".

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")
```



Ex. No. : 6.7 Date:8.5.24

Register No : 231501161 Name : SUBASHINI R

,

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:

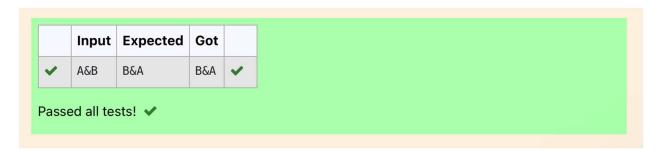
Input Result A&x# x&A#

```
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"</pre>
```



Ex. No. : 6.8 Date:8.5.24

Register No : 231501161 Name : SUBASHINI R

,

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

Program:

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



Ex. No. : 6.9 Date:8.5.24

Register No : 231501161 Name : SUBASHINI R

,

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:

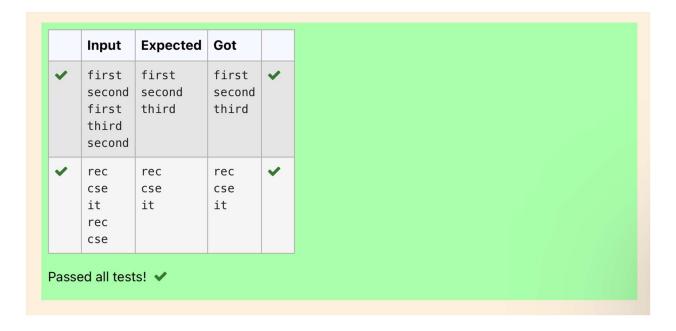
first second first third second

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])
```



Ex. No. : 6.10 Date:8.5.24

Register No: 231501161 Name: SUBASHINI R

,

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:

 $1 \le \text{Length of S} \le 100$

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')
```

