

06 - Strings in Python

For example:

Input	Result
-------	--------

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

3	
---	--

3	
---	--

1	
---	--

Ex. No. : 6.1

Date:

Register No.:

Name:

Count Chars

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

```
s=input()
l=n=c=0
for i in s:
    if(i.isalpha()):
        l+=1
    elif(i.isdecimal()):
        n+=1
    else:
        c+=1
print(l)
print(n)
print(c)
```

Sample Input 1
a2b4c6

Sample Output 1
aabbbbcccccc

Ex. No. : 6.2

Date:

Register No.:

Name:

Decompress the String

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

```
test_string = input()
```

```
res = []
```

```
numbers = []
```

```
temp = ""
```

```
for i in test_string:
```

```
    if i.isnumeric():
```

```
        temp += i
```

```
    else:
```

```
        if temp != "":
```

```
            numbers.append(int(temp))
```

```
            temp = ""
```

```
            res.append(i)
```

```
if temp != "":
```

```
    numbers.append(int(temp))
```

```
for i in range(0,len(res)):
```

```
    print(str(res[i]*numbers[i]),end="")
```

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.

Ex. No. : 6.3

Date:

Register No.:

Name:

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.

```
def common_characters(S1, S2, N):
    common_chars = []
    for char in S1:
        if char in S2 and char not in common_chars:
            common_chars.append(char)
            if len(common_chars) == N:
                break
    return ".join(common_chars)
S1 = input().strip()
S2 = input().strip()
N = int(input().strip())
print(common_characters(S1, S2, N))
```

Sample Input 1
experience
enc

Sample Output 1
xpri

Ex. No. : 6.4

Date:

Register No.:

Name:

Remove Characters

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

Constraints

1 <= string length <= 200

```
s1=input()
s2=input()
s1="".join(i for i in s1 if i not in s2)
print(s1)
```

For example:

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

Ex. No. : 6.5

Date:

Register No.:

Name:

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

```
l=input().lower()
s1=l.split()
for i in s1:
    if(i[::-1]!=i):
        print(i,end=' ')
```

For example:

Input	Result
Wipro Technologies Bangalore	TECHNOLOGIES
Hello World	WORLD
Hello	LESS

Ex. No. : 6.6

Date:

Register No.:

Name:

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.

```
s=input().split()
if(len(s)>=2):
    print(s[1].upper())
else:
    print("LESS")
```

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#	
------	--

Ex. No. : 6.7

Date:

Register No.:

Name:

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.

```
seq =input()
```

```
chars = ['$','%','*','#','^']
```

```
nums = []
```

```
for i in range(len(seq)):
```

```
    if seq[i] not in chars:
```

```
        nums.append(seq[i])
```

```
nums.reverse()
```

```
for j in seq:
```

```
    if j in chars:
```

```
        idx = seq.index(j)
```

```
        nums.insert(idx, j)
```

```
reverse = "".join(nums)
```

```
print(reverse)
```

For example:

Input	Result
Yn	
PYnative	
True	

Ex. No. : 6.8

Date:

Register No.:

Name:

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".

```
def balance(s1,s2):  
    flag=True  
    for i in s1:  
        if i in s2:  
            pass  
        else:  
            flag=False  
    return flag  
s1=input()  
s2=input()  
if balance(s1,s2)==True:  
    print("True")  
else:  
    print("False")
```

Input:

first
second
first
third
second

then your program should display:

Output:

first
second
third

Ex. No. : 6.9

Date:

Register No.:

Name:

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:

```
l=[]
```

```
try:
```

```
    while True:
```

```
        s=input()
```

```
        if s not in l:
```

```
            l.append(s)
```

```
except EOFError:
```

```
    print('\n'.join(l))
```

Example Input/Output 1:

Input:

vijayakumar.r@rajalakshmi.edu.in

Output:

edu.in
rajalakshmi
vijayakumar.r

Ex. No. : 6.10

Date:

Register No.:

Name:

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

```
s=input()
s1=s.find(".")
s2=s.find("@")
print(s[s1+1:])
print(s[s2+1:s1])
print(s[:s2])
```