

## 06 - Strings in Python

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

For example:

Input	Result
rec@123	3
	3
	1

```
S=input()
```

```
L= d = c = 0
```

```
For I in s:
```

```
    If i.isalpha()==True:
```

```
        L=l+1
```

```
    Elif i.isdigit()==True:
```

```
        D = d + 1
```

```
    Else:
```

```
        C = c + 1
```

```
Print(l)
```

```
Print(d)
```

```
Print©
```

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)

Sample Input 1

a2b4c6

Sample Output 1

aabbbbc6cccccc

#in place

```
s=input()
n=len(s)
#s1=""
for i in range(0,n):
    if s[i].isalpha()==True:
        c=""
        for j in range(i+1,n):
            # print(s[j])
            if s[j].isdigit()==True:
                c=c+s[j]
            if s[j].isalpha()==True:
                break
        #print(c)
        s=s+s[i]*int(c)
print(s[n:])
```

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.

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 <= N <= 10

2 <= Length of S1, S2 <= 1000

Example Input/Output 1:

Input:

```
abcde  
cdefghbb  
3
```

Output:

```
bcd
```

Note:

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

```
#string  
s1=input()  
  
s2=input()  
n=int(input())  
l=[]  
for i in s1:  
    if i not in l:  
        l.append(i)  
l1=[]  
for i in l:  
    l1.append(s2.count(i))  
  
for i in range(0,len(l1)):  
    if l1[i]>=1 and n>0:  
        print(l[i],end="")  
        n=n-1
```

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

Sample Input 1

experience

enc

Sample Output 1

xperi

```
#remove characters
```

```
s1=input()
```

```
s2=input()
```

```
s=""
```

```
for i in range(0,len(s1)):
```

```
    if s1[i] not in s2:
```

```
        s=s+s1[i]
```

```
print(s)
```

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

For example:

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

```
S= input().split(" ")
```

```
S_=""
```

```
For x in s[:]:
```

```
    Flag=False
```

```
    X=x.lower()
```

```
    If(x!=x[::-1]):
```

```
        Flag=True
```

```
    If(flag):
```

```
        S_+=x
```

```
        S_+=" "
```

```
    Else:
```

```
        Continue
```

```
print(s_)
```

Ex. No. : 6.6

Date:

Register No.:

Name:

---

## 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

```
#second word
s=input()
z=s.split()
if len(z)==1:
    print("LESS")
else:
    print(z[1].upper())
```

Ex. No. : 6.7

Date:

Register No.:

Name:

---

## Reverse 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#

```
#special characters
```

```
l=[]
```

```
s=input()
```

```
c=0
```

```
n=len(s)
```

```
for i in s:
```

```
    l.append(i)
```

```
z=[]
```

```
#print(l)
```

```
for i in range(len(l)-1,-1,-1):
```

```
    if l[i].isalpha()==True:
```

```
        z.append(l[i])
```

```
        c=c+1
```

```
c=n-c
```

```
#print(z)
```

```
for i in range(0,c):
```

```
    z.append(0)
```

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

```
    if l[i].isalnum()==False:
```

```
        z.insert(i,l[i])
```

```
#print(z)
```

```
for i in z:
```

```
    if i!=0:
```

```
        print(i,end="")
```

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".

For example:

Input Result

~~Yn~~

~~PYnative~~

True

```
#balance
s1=input()
s2=input()
n=len(s1)
c=0
for i in s2:
    if i in s1:
        c=c+1
if c==n:
    print("True")
else:
    print("False")
```



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:

**Input:**

first  
second  
first  
third  
second

then your program should display:

**Output:**

first  
second  
third

#unique

```
l=[]
while True:
    s=input()
    if s=='\n' or s==' ':
        break
    if s not in l:
        l.append(s)
for i in l:
    print(i)
```

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:**

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

Example Input/Output 1:

### **Input:**

vijayakumar.r@rajalakshmi.edu.in

### **Output:**

edu.in  
rajalakshmi  
vijayakumar.r

#email extension

```
s=input()
```

```
a=s.find('@')
```

```
b=s.find('.')
```

```
print(s[b+1:])
```

```
print(s[a+1:b])
```

```
print(s[0:a])
```