

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

```
a=input()
b=0
c=0
d=0
for char in a:
    if(char.isdigit()):
        b+=1
    elif(char.isalpha()):
        c+=1
    else:
        d+=1
print(c)
print(b)
print(d)
```

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)

Solution:

```
s = input()
o = ""
i = 0
while i < len(s):
    char = s[i]
    count = 0
    i += 1
    while i < len(s) and s[i].isdigit():
        count = count * 10 + int(s[i])
        i += 1
    o += char * count
print(o)
```

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.

```
s1=input()
s2=input()
n=int(input())
b=""
```

```
for char in s1:
    if (char in s2) and (char not in b):
        b=b+char
    if len(b)==n:
        break
print(b)
```

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()
```

```
l=len(s1)
```

```
i=0
```

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

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

```
        print(s1[i],end="")
```

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

```
a=input()
f=a.lower()
s=f.split()
c=[]
for i in range(0,len(s)):
    if s[i]!=s[i][::-1]:
        c.append(s[i])
str=" "
s1=str.join(c)
print(s1)
```

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.

```
s=input()
a=[char for char in s if char.isalpha()]
r=a[::-1]
result=""
i=0
for char in s:
    if char.isalpha():
        result+=r[i]
        i+=1
    else:
        result+=char
print(result)
```

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.

```
a=input()
s= a.upper()
b = s.split()
count=0
if len(b)>=2:
    for i in range(0,len(b)):
        if(i==1):
            print(b[i])
            break
        else:
            print("LESS")
```


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".

```
s1=input()
s2=input()
if ((s1 in s2) or (s2 in s1)):
    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:

```
a=[]
while a!=0:
    str=input()
    if str==' ':
        break
    if str not in a:
        a.append(str)
for str in a:
    print(str)
```

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

Solution:

```
s1 = input().strip()
a = s1.find("@")
b = s1.find(".")
u = s1[:a]
c=s1[a+1:b]
e=s1[b+1:]
print(e)
print(c)
print(u)
```