06 - Strings in Python

Ex. No. : 6.1 **Date:**

Register No.: 231001063 **Name: HEMA PRABHA S**

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

```
#chars
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(c)
```

Ex. No. : 6.2 **Date:**

Register No.: 231001063 **Name: HEMA PRABHA S**

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
aabbbbccccccc

```
#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.: 231001063 **Name: HEMA PRABHA S**

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:

abcbde
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.: 231001063 **Name: HEMA PRABHA S**

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

```
#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.: 231001063 **Name: HEMA PRABHA S**

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

```
#palindrome
s=input()
s=s.lower()
z=s.split()

for i in z:
    if i!=i[::-1]:
        print(i,end=' ')
```

Ex. No. : 6.6 **Date:**

Register No.: 231001063 **Name: HEMA PRABHA S**

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.: 231001063 **Name: HEMA PRABHA S**

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.: 231001063 **Name: HEMA PRABHA S**

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.: 231001063 **Name: HEMA PRABHA S**

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)
```