06 - Strings in Python

For example:

Input Result
rec@123
3
3
1

<u>Count Chars</u>

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

```python
s=input()
n=len(s)
count=0
for i in s:
    if i.isalpha():
        count +=1

print(count)
result1=0
for j in s:
    if j.isnumeric():
        result1+=1

print(result1)
result3=n-count-result1
print(result3)
```

Sample Input 1
a2b4c6

Sample Output 1
aabbbbcccccc

Decompress the String

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

```python
s = list(input().strip())
i = 0

while i < len(s):
    if s[i].isdigit():
        j = i
        while j < len(s) and s[j].isdigit():
            j += 1
        count = int(''.join(s[i:j]))
        char = s[i - 1]
        s[i - 1] = char * count
        del s[i:j]
        i += count - 1
    else:
        i += 1

print(''.join(s))
```

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.

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.

```python
S1 = input().strip()
S2 = input().strip()
N = int(input().strip())

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

print(''.join(common_chars))
```

Sample Input 1
experience
enc

Sample Output 1
xpri

Remove Characters

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

Constraints
1<= string length <= 200

```
n1=input()
n2=list(input())
str1="
for i in n1:
    if i not in n2:
        str1+=i
print(str1)
```

For example:

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

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

```
n=input()
n=n.lower()
a=n.split()
c=[]
for i in range(len(a)):
   c.append("")
   r=len(a[i])
   for j in range(r):
      c[i]+=a[i][r-j-1]
   if(c[i]!=a[i]):
      c[i]=0
for i in range(len(a)):
   if(c[i]==0):
      print(a[i],end=' ')
```

For example:

Input Result
Wipro Technologies Bangalore
TECHNOLOGIES
Hello World
WORLD
Hello
LESS

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()
b=a.split(" ")
if(len(b)>1):
    print(b[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#

<u>Revers String</u>

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.
For example:

Input Result
Yn
PYnative
True

```
s = input().strip()
s_list = list(s)
left = 0
right = len(s_list) - 1
while left < right:
    if not s_list[left].isalpha():
        left += 1
    elif not s_list[right].isalpha():
        right -= 1
    else:
        s_list[left], s_list[right] = s_list[right], s_list[left]
        left += 1
        right -= 1
result = ''.join(s_list)
print(result)
```

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".

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

Input:
first
second
first
third
second

then your program should display:

Output:
first
second
third

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:

```
s=[]
b=0
for i in range (5):
    a=input()
    if (a not in s):
        s.append(a);
        b=b+1
for i in range (b):
    print(s[i])
```

Example Input/Output 1:

Input:

vijayakumar.r@rajalakshmi.edu.in

Output:

edu.in
rajalakshmi
vijayakumar.r

<u>Username Domain Extension</u>

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()

a=s.find(".")

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

b=s.find("@")

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

print(s[0:b:1])