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()
alp=0
dig=0
count=0
for i in range(0,len(a)):
  if(a[i].isalpha()):
    alp=alp+1
  elif(a[i].isdigit()):
    dig=dig+1
  else:
    count=count+1
print(alp)
print(dig)
print(count)
```

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)

```
def expand_string(s):
    result = ""
    for i in range(0, len(s), 2):
        char = s[i]
        count = int(s[i + 1])
        result += char * count
    return result

input_string = input()
expanded_string = expand_string(input_string)
print(expanded_string)
```

**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 find_common_characters(S1, S2, N):
    common_chars = [char for char in S1 if char in S2]
    return ".join(common_chars[:N])

S1 = input()
S2 = input()
N = int(input())

result = find_common_characters(S1, S2, N)
print(result)
```

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

```
a=input()

b=input()

for i in range(0,len(b)):

   for j in range(0,len(a)):

      if(b[i]==a[j]):

         a=a.replace(a[j],' ')

for i in range(0,len(a)):

   if(a[i]!=' '):

      print(a[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

```
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=' ')
```

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.

```
def get_second_word_uppercase(sentence):
    words = sentence.split()
    if len(words) < 2:
        return "LESS"
    else:
        return words[1].upper()

sentence = input().strip()

result = get_second_word_uppercase(sentence)
print(result)
```

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.

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

For example:

Input Result

Yn

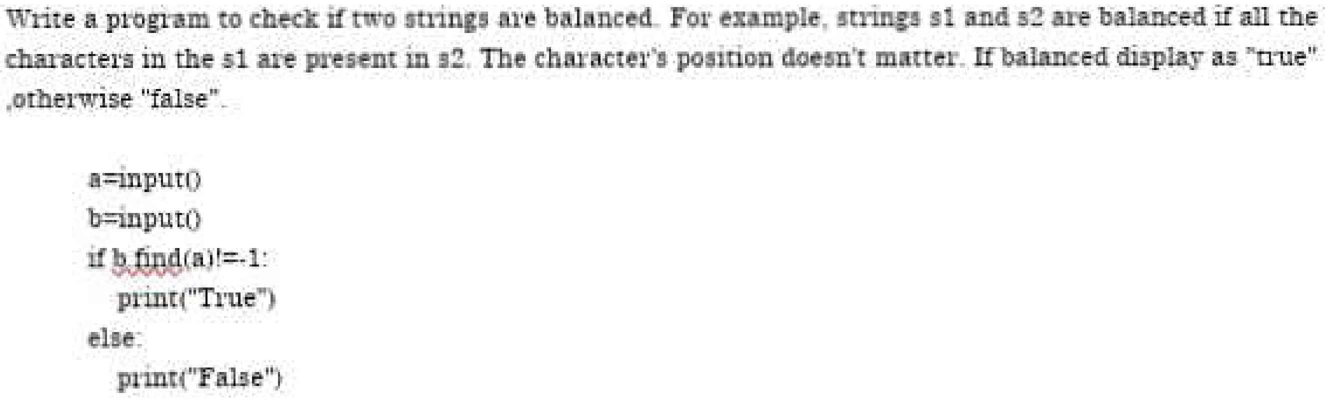
PYnative

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".

```
a=input()
b=input()
if b.find(a)!=-1:
   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:

```
s=[]
for i in range(5):
   a=input()
   if a not in s:
      s.append(a)
for i in s:
   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 <= Length of S <= 100

```
a=input()
b=a.split()
c=len(b[0])
d=0
for i in range(len(b)):
   if c<len(b[i]):
       c=len(b[i])
       d=i
print(b[d],c,sep='\n')
```