1. Find if a String2 is substring of String1. If it is, return the index of the first occurrence. else return -1.

**Sample Input 1**

thistest123string

123

**Sample Output 1**

8

s1=input().strip()

s2=input().strip()

f=False

i=-1

for i in range(len(s1)-len(s2)+1):

if s1[i:i+len(s2)]==s2:

f=True

i=i

break

if f:

print(i)

else:

print(-1)

2. 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=n=c=0

for i in s:

if(i.isalpha()):

l+=1

elif(i.isdecimal()):

n+=1

else:

c+=1

print(l)

print(n)

print(c)

3. 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.

s1 = input().strip()

s2 = input().strip()

n = int(input().strip())

common\_chars = set(s1) & set(s2)

result = []

for char in s1:

if char in common\_chars:

result.append(char)

common\_chars.remove(char)

if len(result) == n:

break

print(''.join(result))

4. 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:

first

second

first

third

second

then your program should display:

first

second

third

l=[]

try:

while True:

s=input()

if s not in l:

l.append(s)

except EOFError:

print('\n'.join(l))

5. 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()

for i in a.split():

i=i.lower()

if i!=i[::-1]:

print(i,end=' ')

6. 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

s1=list(input())

s2=list(input())

for i in s2:

while i in s1:

s1.remove(i)

print(''.join(s1))

7. Given a string S which is of the format [USERNAME@DOMAIN.EXTENSION](mailto: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

**Example Input/Output 1:**

Input:

[abcd@gmail.com](mailto:abcd@gmail.com)

Output:

com  
gmail  
abcd

**For example:**

|  |  |
| --- | --- |
| **Input** | **Result** |
| [arvijayakumar@rajalakshmi.edu.in](mailto:arvijayakumar@rajalakshmi.edu.in) | edu.in rajalakshmi arvijayakumar |

s=input()

f=s.find('@')

r=s.find('.')

print(s[r+1:])

print(s[f+1:r])

print(s[:f])

8. 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 |

s=input().split()

if(len(s)>=2):

print(s[1].upper())

else:

print("LESS")

9. Assume that the given string has enough memory.

Don't use any extra space(IN-PLACE)

**Sample Input 1**

a2b4c6

**Sample Output 1**

aabbbbcccccc

s=input()

temp=0

char=''

for i in s:

if i.isalpha():

print(char\*temp,end='')

temp=0

char=i

else:

temp=temp\*10+int(i)

print(char\*temp,end='')

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 |

print(input() in input())