1. Consider the below words as key words and check the given input is key word or not.

keywords: {break, case, continue, default, defer, else, for, func, goto, if, map, range, return, struct, type, var}

Input format:

Take string as an input from stdin.

Output format:

Print the word is key word or not.

Example Input:

break

Output:

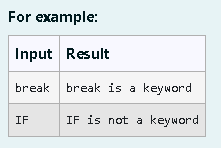
break is a keyword

Example Input:

IF

Output:

IF is not a keyword



Program:

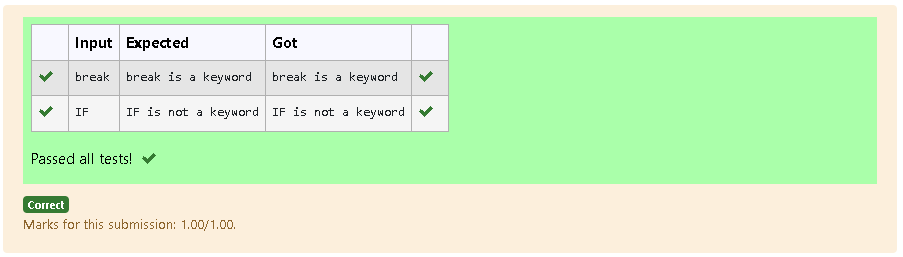
a=input()

if(a=='break' or a=='case'or a=='continue' or a=='default' or a=='defer' or a=='else' or a=='for' or a=='func' or a=='goto' or a=='if' or a=='map' or a=='range' or a=='return' or a=='struct' or a=='type' or a=='var'):

print(a,'is a keyword')

else:

print(a,'is not a keyword')



2. 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.

Program:

s1=input().strip()

s2=input().strip()

n=int(input())

r=''

c=0

for char in s1:

if char in s2 and char not in r:

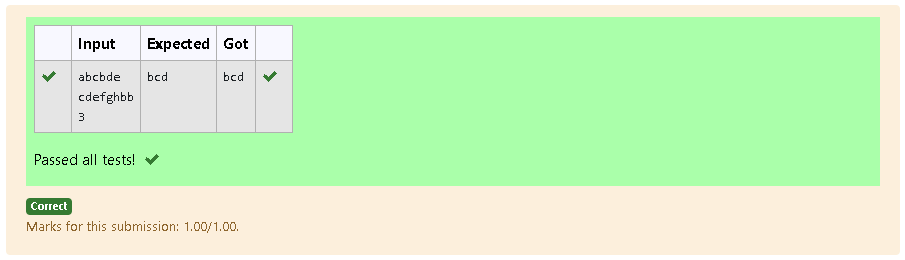
r+=char

c+=1

if c==n:

break

print(r)



3. 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

Program:

a=input()

b=input()

a=list(a)

b=list(b)

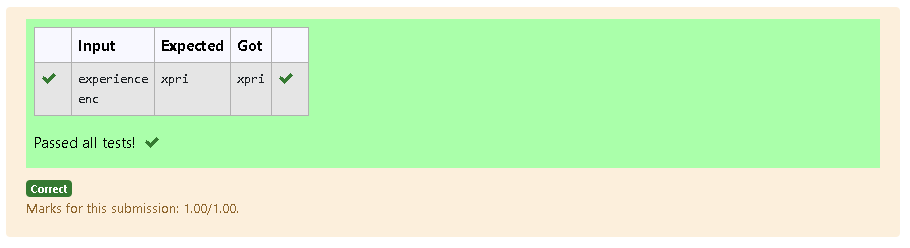
s=''

for i in a:

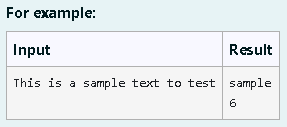
if(i not in b):

s+=i

print(s)



4.  Write a python to read a sentence and print its longest word and its length



Program:

a=input()

a=a.split()

m=''

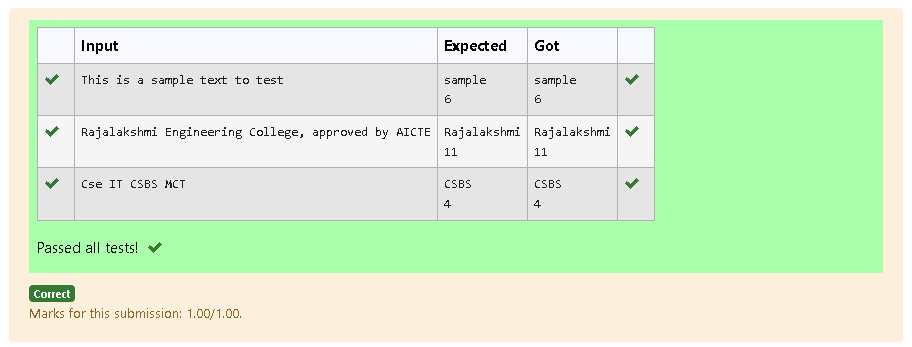
for i in a:

if len(i)>len(m):

m=i

print(m)

print(len(m))



5. 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

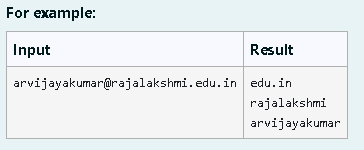
**Example Input/Output 1:**

Input:

abcd@gmail.com

Output:

com  
gmail  
abcd



Program:

s=input().strip()

at\_index=s.find("@")

dot\_index=s.find(".")

extension=s[dot\_index+1:]

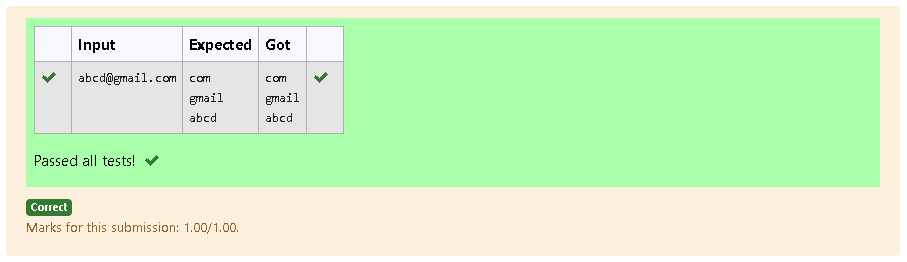
domain=s[at\_index+1:dot\_index]

username=s[:at\_index]

print(extension)

print(domain)

print(username)



6.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

Program:

ws=[]

while True:

w=input().strip()

if w=="":

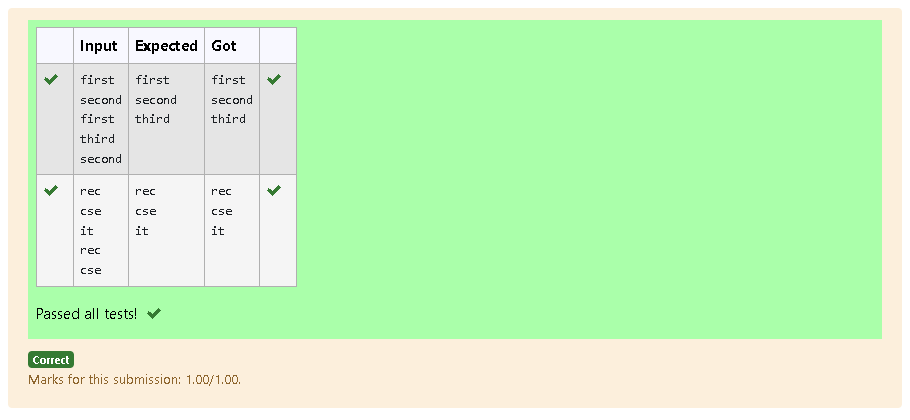
break

if w not in ws:

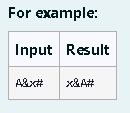
ws.append(w)

for w in ws:

print(w)



7. **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".



Program:

s=input().strip()

ch=list(s)

left,right=0,len(ch)-1

while left<right:

if not ch[left].isalpha():

left+=1

elif not ch[right].isalpha():

right-=1

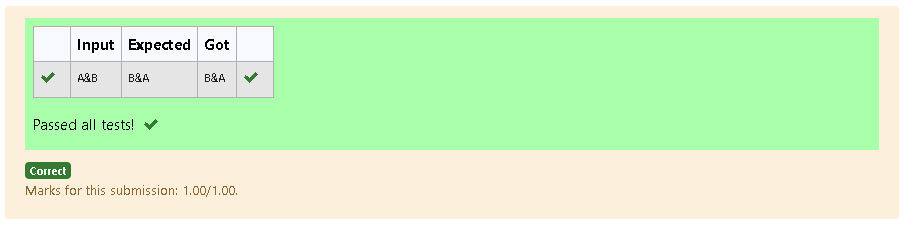
else:

ch[left],ch[right]=ch[right],ch[left]

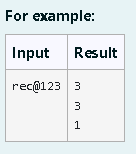
left+=1

right-=1

print("".join(ch))



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



Program:

a=input()

l=0

d=0

ss=0

for i in a:

if i.isalpha():

l+=1

elif i.isnumeric():

d+=1

else:

ss+=1

print(l)

print(d)

print(ss)



9. 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

Program:

a=input()

b=input()

# c=0

# for i in a:

# if(i.isalpha()):

# c+=1

# else:

# break

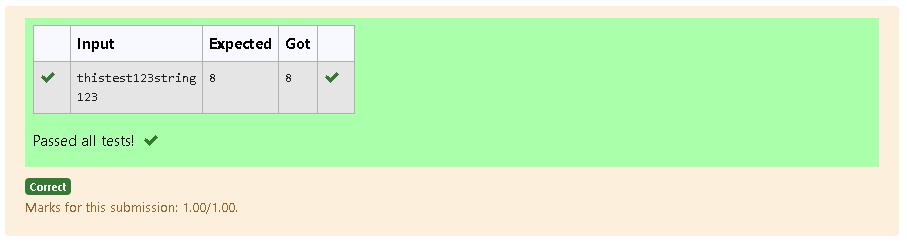
if b in a:

print(a.index(b))

else:

print(-1)

# print(c)



10. Given a string S, which contains several words, print the count C of the words whose length is atleast L. (You can include punctuation marks like comma, full stop also as part of the word length. Space alone must be ignored)

**Input Format:**

The first line contains S.  
The second line contains L.

**Output Format:**

The first line contains C

**Boundary Conditions:**

2 <= Length of S <= 1000

**Example Input/Output 1:**

Input:

During and after Kenyattas inauguration police elsewhere in the capital, Nairobi, tried to stop the opposition from holding peaceful demonstrations.  
5

Output:

13

Explanation:

The words of minimum length 5 are  
During  
after  
Kenyattas  
inauguration  
police  
elsewhere  
capital,  
Nairobi,  
tried  
opposition  
holding  
peaceful  
demonstrations.

Program:

s=input().strip()

l=int(input())

w=s.split()

c=0

for ws in w:

if len(ws)>=l:

c+=1

print(c)

