1. String should contain only the words are not palindrome.

**Sample Input 1**

Malayalam is my mother tongue

**Sample Output 1**

is my mother tongue

Answer

a=input()

for i in a.split():

i=i.lower()

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

print(i,end=' ')

2.Robert  is having 2 strings consist of uppercase & lowercase english letters. Now he want to compare those two strings lexicographically. The letters' case does not matter, that is an uppercase letter is considered equivalent to the corresponding lowercase letter.

### Input

The first line contains **T**. Then **T** test cases follow.

Each test case contains a two lines contains a string. The strings' lengths range from 1 to 100 inclusive. It is guaranteed that the strings are of the same length and also consist of uppercase and lowercase Latin letters.

### Output

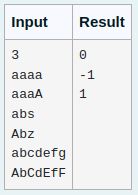
If the first string is less than the second one, print "-1".  
If the second string is less than the first one, print "1".  
If the strings are equal, print "0".  
Note that the letters' case is not taken into consideration when the strings are compared.

### Constraints

**1**≤**T**≤**50**

**String length**≤**100**

For example:



Answer

n=int(input())

flag=0

for i in range(n):

s1=input().lower()

s2=input().lower()

for j in range(len(s1)):

if(s1[j]<s2[j]):

print(-1)

flag=1

break

elif(s1[j]>s2[j]):

print(1)

flag=1

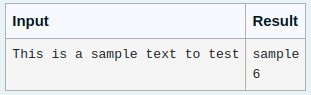
break

if(flag==0):

print(0)

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

For example:



Answer

sentence = input()

words = sentence.split()

longest\_word = ""

longest\_length = 0

for word in words:

if len(word) > longest\_length:

longest\_word = word

longest\_length = len(word)

print(longest\_word)

print(longest\_length)

4.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

Answer

s1=list(input())

s2=list(input())

for i in s2:

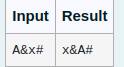
while i in s1:

s1.remove(i)

print(''.join(s1))

5.**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:



Answer

s=list(input())

i=0

j=len(s)-1

while(i<=j):

if(s[i].isalpha()):

while(not s[j].isalpha()):

j-=1

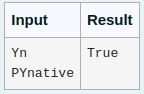
s[i],s[j]=s[j],s[i]

i+=1

print(''.join(s))

6. 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:



Answer

print(input() in input())

7. n 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

Answer

l=[]

try:

while True:

s=input()

if s not in l:

l.append(s)

except EOFError:

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

8.Assume that the given string has enough memory.

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

**Sample Input 1**

a2b4c6

**Sample Output 1**

aabbbbcccccc

Answer

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='')

9. 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.

Answer

words=input().split(' ')

k=int(input())

c=0

for i in words:

if len(i)>=k:

c+=1

print(c)

10. 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.

Answer

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))