Ex. No. : 5.1 **Date: 17.04.24**

Register No.: 230701385 **Name: S. Vishwak**

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".

Input	Result
Yn PYnative	True

For example:

Program:

```
s1=input()
s2=input()
count=0
for i in s1:
    if(i in s2):
        count+=1
if(count==len(s1)):
    print(True)
else:
    print(False)
```

	Input	Expected	Got	
✔	Yn PYnative	True	True	✔
✔	Ynf PYnative	False	False	✔

Ex. No. : 5.2 **Date: 17.04.24**

Register No.: 230701385 **Name: S. Vishwak**

.

Decompress the String

Assume that the given string has enough memory. Don't use any extra space(IN-PLACE)
Sample Input 1
a2b4c6

Sample Output 1
aabbbbccccccc

Program:

```
str1=input()
alpha=[]
num=[]
current_num=""
for ch in str1:
    if ch.isalpha():
        if current_num:
            num.append(current_num)
            current_num=""
        alpha.append(ch)
    elif ch.isdigit():
        current_num+=ch
if current_num:
    num.append(current_num)
for i in range(0,len(alpha)):
    print(int(num[i])*alpha[i],end='')
```

	Input	Expected	Got	
✔	a2b4c6	aabbbbcccccc	aabbbbcccccc	✔
✔	a12b3d4	aaaaaaaaaaaabbbdddd	aaaaaaaaaaaabbbdddd	✔

Ex. No. : 5.3 **Date: 17.04.24**

Register No.: 230701385 **Name: S. Vishwak**

.

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.
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:

```
str1=input()
str2=input()
n=int(input())
list1=[]

for i in str1:
    count=0
    for j in str2:
        if(i==j):
            count+=1
    if(count>0 and i not in list1):
        list1.append(i)

for i in range(0,len(list1)-1):
    print(list1[i],end='')
```

	Input	Expected	Got	
✔	abcbde cdefghbb 3	bcd	bcd	✔

Ex. No. : 5.4 **Date: 17.04.24**

Register No.: 230701385 **Name: S. Vishwak**

.

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

Example Input/Output 1:
Input:
vijayakumar.r@rajalakshmi.edu.in
Output:
edu.in
rajalakshmi
vijayakumar.r

Program:

```
str1=input()
domain=""
ext=""
name,rest=str1.split('@')
for i in range(0,len(rest)):
    if rest[i]=='.':
        break
for a in range(0,i):
    domain+=rest[a]
```

```
for b in range(i+1,len(rest)):
    ext+=rest[b]
print(ext)
print(domain)
print(name)
```

	Input	Expected	Got	
✔	abcd@gmail.com	com gmail abcd	com gmail abcd	✔

Ex. No. : 5.5 **Date: 17.04.24**

Register No.: 230701385 **Name: S. Vishwak**

.

Count Chars

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

Program:

```
str1=input()
alph=0
num=0
spc=0
for i in str1:
    if(i.isalpha()):
        alph+=1
    elif(i.isnumeric()):
        num+=1
    else:
        spc+=1
print(alph,num,spc,sep='\n')
```


	Input	Expected	Got	
✔	rec@123	3 3 1	3 3 1	✔
✔	P@#yn26at^&i5ve	8 3 4	8 3 4	✔
✔	abc@12&	3 2 2	3 2 2	✔

Ex. No. : 5.6 **Date: 17.04.24**

Register No.: 230701385 **Name: S. Vishwak**

.

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:

```
str1=input()
str2=input()
if(str2 in str1):
    print(str1.index(str2))
else:
    print(-1)
```

Ex. No. : 5.7 **Date: 17.04.24**

Register No.: 230701385 **Name: S. Vishwak**

.

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

Program:

```
str1=input()
str2=str1.split()
if(len(str2)>1):
    print(str2[1].upper())
else:
    print("LESS")
```


Ex. No. : 5.8 **Date: 17.04.24**

Register No.: 230701385 **Name: S. Vishwak**

.

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

Program:

```
str1=input()
str1=str1.lower()
str2=str1.split()
for i in str2:
    if(i==i[::-1]):
        continue
    else:
        print(i,end=' ')
```

	Input	Expected	Got	
✔	Malayalam is my mother tongue	is my mother tongue	is my mother tongue	✔

Ex. No. : 5.9 **Date: 17.04.24**

Register No.: 230701385 **Name: S. Vishwak**

.

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:

Input	Result
3 aaaa aaaA abs Abz abcdefg AbCdEfF	0 -1 1

Program:

```
n=int(input())
```

```
while(n!=0):
    str1=input()
    str1=str1.lower()
    str2=input()
    str2=str2.lower()
    if(str1==str2):
        print(0)
    elif(str1>str2):
        print(1)
    else:
        print(-1)
    n-=1
```

Ex. No. : 5.10 **Date: 17.04.24**

Register No.: 230701385 **Name: S. Vishwak**

.

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:

```
str1=input()
n=int(input())
count=0
str2=str1.split()
for i in str2:
    if(len(i)>=n):
        count+=1

    else:
        continue
print(count)
```