WEEK -5 PYTHON PROGRAMMING

1.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

SOLUTION:

```
a=input()
c1=0
c2=0
c3=0
for  i in a:
    if i>='a'and i<='z':
        c1=c1+1
    elif i>='A'and i<='Z':
        c1=c1+1
    elif i>='0' and i<='9':
        c2=c2+1
        #print(i)
    else:
        c3=c3+1
print(c1)
print(c2)
print(c3)
```

2.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`

SOLUTION:

```
a=input()
j=0
c=0
l=len(a)
#print(l)
for i in a:
    if i==' ':
        #print(j)
        j=c+1
        break
    c=c+1
#print(j)
while a[j]!=' ':
   # print(j)
    if j==0:
        print('LESS')
        break
    print(a[j].upper(),end='')
    j=j+1
    if  j==l:
        break
#print(j)
```

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

For example:

Input	Result

`This is a sample text to test`	`sampl`
	`e`
	`6`

SOLUTION:

```
a=input()
li=a.split(' ')
#print(li)
l=len(li)
b=len(li[0])
c=li[0]
for j in range (0,l,1):
    if b<len(li[j]):
        b=len(li[j])
        c=li[j]
print(c)
print(b)
```

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

SOLUTION:

```
l=[]
while True:
    s=input()
    if s=='\n' or s==' ':
        break
    if s not in l:
        l.append(s)
for i in l:
    print(i)
```

5.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.

SOLUTION:

```
a=input()
b=input()
n=int(input())
a=set(a)
```

```
a=list(a)
a.sort()
b=set(b)
b=list(b)
b.sort()
#print(b)
#print(a)
c=0
for i in a:
    for j in b:
        if i==j:
            print(i,end='')
            c=c+1
    if c==n:
        break
```

6.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 abcdef g AbCdEf F	0 -1 1

SOLUTION:

```
t= int(input())
i=0
for i in range(0,t):
    a= input()
    b=input()
    a=a.lower()
    b=b.lower()
   # print(a)
   # print(b)
    if a>b:
        print('1')
    elif a<b:
        print('-1')
    elif a==b:
        print('0')
```

7.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

SOLUTION:

```
s1=input()
s2=input()
s1=list(s1)
s2=list(s2)
#print(s1,s2)
for j in s2:
    for i in s1:
        if j==i:
            s1.remove(j)
for i in s1:
    print(i,end='')
```

8.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

For example:

Input	Result
`break`	`break is a keyword`

IF	IF is not a keyword

SOLUTION:

```
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=='goto'or a=='if' or a=='map' or a=='return' or a=='range'or a=='struct' or a=='type' or a=='var'):
    print(a,'is a keyword')
else:
    print(a,'is not a keyword')
```

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

SOLUTION:

```
a=input()
b=input()
flag=False
if b in a:
    n=a.index(b)
    flag=True
if flag:
    print(n)
else:
    print(-1)
```

10. 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:

Input	Result
A&x#	x&A#

SOLUTION:

```
a=[]
d=input()
c=0
n=len(d)
for i in d:
    a.append(i)
z=[]
for i in range(len(a)-1,-1,-1):
    if a[i].isalpha()==True:
        z.append(a[i])
        c=c+1
c=n-c
for i in range(0,c):
    z.append(0)
for i in range(0,len(d)):
    if a[i].isalnum()==False:
        z.insert(i,a[i])
for i in z:
    if i!=0:
        print(i,end='')
```