WEEK 5:

1. 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 52.

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

Length of S1, S2 < 1000

Example Input/Output 1:

**Input**:

abcbde

cdefghbb

3

**Output**

bcd

Note:

boccurs twice in common but must be printed only once.

**Program:**

s1 = input()

s2 = input()

n = int(input())

common = []

for char in s1:

if char in s2 and char not in common:

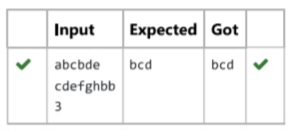
common.append(char)

if len(common) == n:

break

print(''.join(common))

**Output:**

****

2. 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:**

s = input()

words = s.split()

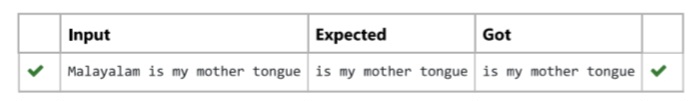
for word in words:

word = word.lower()

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

print(word, end=" ")

**Output:**

****

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

Example:

Input:

rec@123

Output:

3

3

1

**Program:**

string = input()

letter = 0

digit = 0

special = 0

for char in string:

if char.isalpha():

letter += 1

elif char.isdigit():

digit += 1

else:

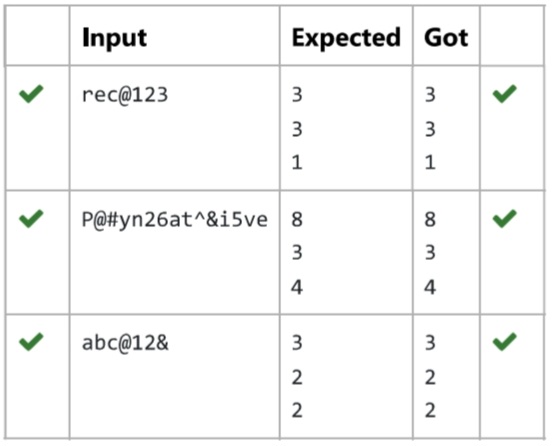
special += 1

print(letter)

print(digit)

print(special)

**Output:**

****

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

Sample Input 1

thistest123string

123

Sample Output 1

8

**Program:**

s1 = input()

s2 = input()

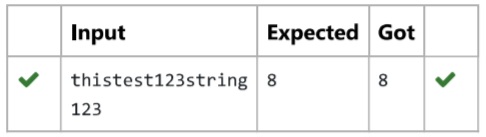
if s2 in s1:

print(s1.index(s2))

else:

print(-1)

**Output:**

****

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

For example:

Input:

This is a sample text to test

Output:

sample

6

**Program:**

sentence = input().split()

longest\_word = ""

length = 0

for word in sentence:

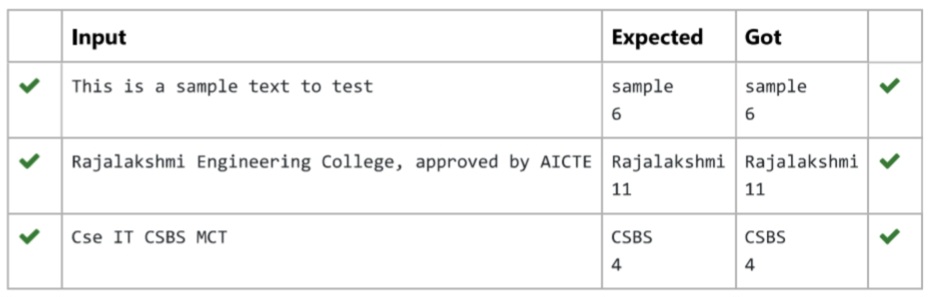
if len(word) > length:

longest\_word = word

length = len(word)

print(longest\_word)

print(length)

**Output:**

6. 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:

Sanple Input:

A&x#

Sample output

x&A#

**Program:**

s = input()

letters = []

for i in s:

if i.isalpha():

letters.append(i)

letters.reverse()

r = ''

index = 0

for i in s:

if i.isalpha():

r += letters[index]

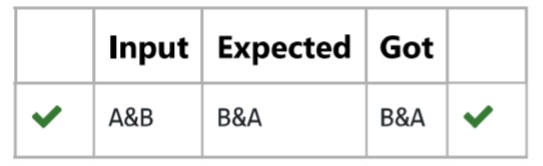
index += 1

else:

r += i

print(r)

**Output:**

****

7. 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: Iake 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:**

word = input()

c = 0

keywords = ['break', 'case', 'continue', 'default', 'defer', 'else', 'for', 'func', 'goto', 'if']

for i in range(len(keywords)):

if keywords[i] == word:

c = c + 1

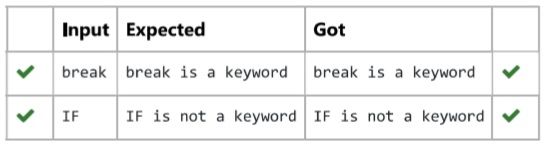
if c == 1:

print(word, "is a keyword")

else:

print(word, "is not a keyword")

**Output:**



8.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:

Yn

PYnative

Output:

True

**Program:**

s1 = input()

s2 = input()

if s1 in s2:

print("True")

else:

print("False")

**Output:**

****

9.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:

Input:

First

Second

First

Third

Second

Then your program should display:

Output:

First

Second

Third

**Program:**

words = []

while True:

word = input().strip()

if word == "":

break

words.append(word)

result = set()

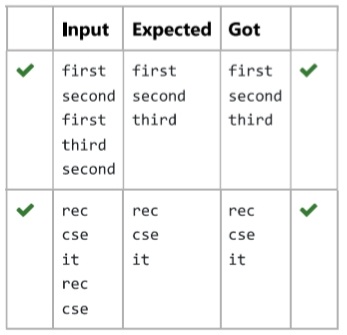
for word in words:

if word not in result:

print(word)

result.add(word)

**Output:**

****

10.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”.

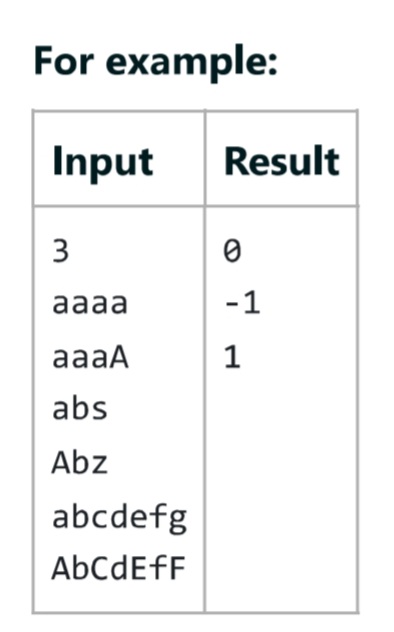
If the strings are equal, print “0”.

Note that the letters’ case is not taken into consideration when the strings are compared.

Constraints

1<=T<=50

String length:100



**Program:**

T = int(input())

for \_ in range(T):

str1 = input().lower()

str2 = input().lower()

if str1 < str2:

print("-1")

elif str1 > str2:

print("1")

else:

print("0")

**Output:**

