Week 5 –

1. Given a string S which is of the format USERNAME@DOMAIN.EXTENSION, the program must print the EXTENSION, DOMAIN, USERNAME in the reverse order.

Example Input/Output 1:

Input: abcd@gmail.com

Output:

Com

gmail

abcd

PROGRAM:

s=input()

at\_index=s.index('@')

dot\_index=s.index('.')

username=s[:at\_index]

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

extension=s[dot\_index+1:]

print(extension)

print(domain)

print(username)

OUTPUT:



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

Sample Input1

Malayalam is my mother tongue

Sample Output1 1

is my mother tongue

Program:

s=input()

words=s.split()

x=’’

for word in words:

word=word.lower()

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

print(word,end=" ")

OUTPUT:



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

PROGRAM:

sentence = input()

words = sentence.split()

longest\_word = ""

longest\_length = 0

for word in words:

if len(word) > longest\_length:

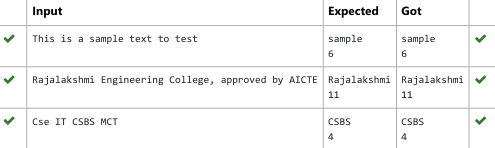
longest\_word = word

longest\_length = len(word)

print(longest\_word)

print(longest\_length)

OUTPUT:



4. 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:

a=input()

b=input()

n=int(input())

bset=set(b)

cc=[]

c=0

for char in a:

if char in bset and char not in cc:

cc.append(char)

c=c+1

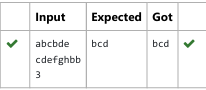
if c==n:

break

x=''.join(cc)

print(x)

OUTPUT:



5. Find if a String2 is substring of String1. If it is, return the index of the irst occurrence. else return -1.

Sample Input 1

thistest123string

123

Sample Output 1

8

PROGRAM:

a=input()

b=input()

if b in a:

print(a.index(b))

else:

print("-1")

OUTPUT:



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:

i=1

x=[]

while(i==1):

a=input()

if(a==" "):

i=0

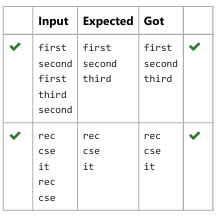
elif a!=" " and a not in x:

x.append(a)

for i in x:

print(i)

OUTPUT:



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()

l=[]

for i in s:

if(i.isalpha()):

l.append(i)

l.reverse()

r=''

index=0

for i in s:

if(i.isalpha()):

r+=l[index]

index+=1

else:

r+=i

print(r)

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".

PROGRAM:

a=input()

b=input()

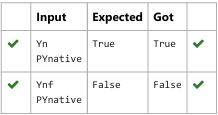
if a in b:

print("True")

else:

print("False")

OUTPUT:



9. 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.

PROGRAM:

a=input()

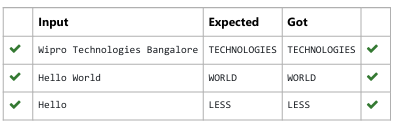
x=a.split()

if(len(x)>=2):

print(x[1].upper())

else:

print("LESS")

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:

def compare\_strings(str1, str2):

str1 = str1.lower()

str2 = str2.lower()

if str1 < str2:

return "-1"

elif str1 > str2:

return "1"

else:

return "0"

T = int(input(""))

for \_ in range(T):

str1 = input().strip()

str2 = input().strip()

print(compare\_strings(str1, str2))

OUTPUT:

