vl dharshini
231501038
aiml a

1) **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:

```
def reverse_string(s):
    s = list(s)
    l, r = 0, len(s) - 1
    while l < r:
        if not s[l].isalpha():
            l += 1
        elif not s[r].isalpha():
            r -= 1
        else:
            s[l], s[r] = s[r], s[l]
            l += 1
            r -= 1
    return ''.join(s)

# Test Cases
input_str=input()
ouput_str=reverse_string(input_str)
print(ouput_str)
```

OUTPUT:

	Input	Expected	Got	
✔	A&B	B&A	B&A	✔

Passed all tests! ✔

2) 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:

abcd@gmail.com

Output:

com
gmail
abcd

PROGRAM:

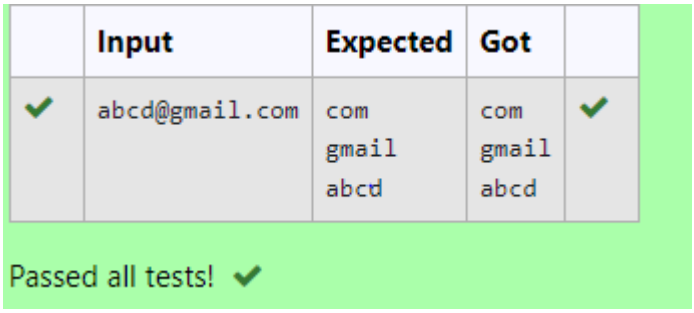
```
email=input()
dot=email.index('.')
at=email.index('@')
# dot=email.index('.')
```

```
print(email[dot+1:])

print(email[at+1:dot])

print(email[:at])
```

OUTPUT:



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

Passed all tests! ✔

3) 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:

```
def are_strings_balanced(s1, s2):
    return set(s1).issubset(set(s2))

s1 = input()

s2 = input()

print(are_strings_balanced(s1, s2))
```

OUTPUT:

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

Passed all tests! ✔

4) Write a python to read a sentence and print its longest word and its length

PROGRAM:

```
sentence = input()
words = sentence.split()
longest_word = max(words, key=len)
print( longest_word)
print( len(longest_word))
```

OUTPUT:

	Input	Expected	Got	
✔	This is a sample text to test	sample 6	sample 6	✔
✔	Rajalakshmi Engineering College, approved by AICTE	Rajalakshmi 11	Rajalakshmi 11	✔
✔	Cse IT CSBS MCT	CSBS 4	CSBS 4	✔

Passed all tests! ✔

5) 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:

```
def extract_second_word(sentence):
    words = sentence.split()
    if len(words) < 2:
        return "LESS"
    return words[1].upper()

# Test the function
input_sentence = input()
result = extract_second_word(input_sentence)
print(result)
```

OUTPUT:

	Input	Expected	Got	
✔	Wipro Technologies Bangalore	TECHNOLOGIES	TECHNOLOGIES	✔
✔	Hello World	WORLD	WORLD	✔
✔	Hello	LESS	LESS	✔

Passed all tests! ✔

6) 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
```

PROGRAM:

```
s1 = input()
s2 = input()

result = ''.join([char for char in s1 if char not in s2])

print(result)
```

OUTPUT:

	Input	Expected	Got	
✔	experience enc	xpri	xpri	✔

Passed all tests! ✔

7) 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:

```
a=input().lower()
a=a.split()
s=[]
for i in a:
    if i!=i[::-1]:
        s.append(i)
print(' '.join(s))
```

OUTPUT:

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

Passed all tests! ✔

8) 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

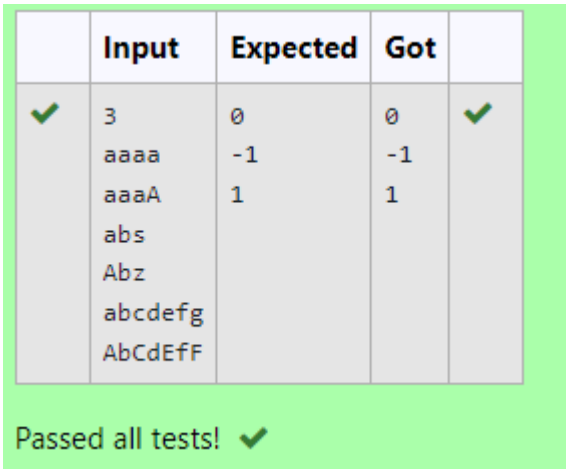
PROGRAM:

```
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:



	Input	Expected	Got	
✔	3 aaaa aaaA abs Abz abcdefg AbCdEfF	0 -1 1	0 -1 1	✔

Passed all tests! ✔

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

PROGRAM:

```
def count_chars(input_string):
    letters = 0
    digits = 0
    special_symbols = 0

    for char in input_string:
        if char.isalpha():
```

```
            letters += 1
        elif char.isdigit():
            digits += 1
        else:
            special_symbols += 1

    return letters, digits, special_symbols

input_string = input()
letters, digits, special_symbols = count_chars(input_string)

print(letters)
print(digits)
print(special_symbols)
```

OUTPUT:

	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	✔

Passed all tests! ✔

10) 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:

```
def find_substring(string1, string2):
    if string2 in string1:
        return string1.index(string2)
    else:
        return -1

# Sample Input
string1 = input()
string2 = input()

# Sample Output
print(find_substring(string1, string2))
```

OUTPUT:

	Input	Expected	Got	
✔	thistest123string 123	8	8	✔

Passed all tests! ✔