[Dashboard](http://118.185.187.137/moodle/my/) / [My courses](http://118.185.187.137/moodle/my/courses.php) / [CS23221-PPL-2023](http://118.185.187.137/moodle/course/view.php?id=148) / [Experiments based on Strings and its operations.](http://118.185.187.137/moodle/course/view.php?id=148&section-5) / [Week5\_Coding](http://118.185.187.137/moodle/mod/quiz/view.php?id=1067)

**Started on** Saturday, 27 April 2024, 1:20 PM

**State** Finished

**Completed on** Monday, 13 May 2024, 11:19 AM

**Time taken** 15 days 21 hours

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

**Answer:** (penalty regime: 0 %)

|  |  |
| --- | --- |
| 1  2  3  4 ▼  5 ▼  6  7 | a=input() b=input() p=""  for char in a:  if char not in b: p=p+char  print(p) |
|  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | experience enc | xpri | xpri |  |

Passed all tests! 

Question **2**

Correct

Marked out of 1.00

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.

**Answer:** (penalty regime: 0 %)

|  |  |
| --- | --- |
| 1  2  3  4  5 ▼  6 ▼  7  8  9 | a=input() b=int(input()) c=a.split() count=0  for i in c:  if(len(i)>=b): count+=1  print(count) |
|  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | During and after Kenyattas inauguration police elsewhere in the capital, Nairobi, tried to stop the opposition from holding peaceful demonstrations.  5 | 13 | 13 |  |

Passed all tests! 

Question **3**

Correct

Marked out of 1.00

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

**Answer:** (penalty regime: 0 %)

|  |  |
| --- | --- |
| 1  2 ▼  3  4 ▼  5  6 ▼  7  8 ▼  9 | ww=[]  while True:  w=input() if w==" ":  break  elif w not in ww: ww.append(w)  for w in ww:  print(w) |
|  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | first | first | first |  |
| second | second | second |
| first | third | third |
| third |  |  |
| second |  |  |
|  | rec | rec | rec |  |
| cse | cse | cse |
| it | it | it |
| rec |  |  |
| cse |  |  |

Passed all tests! 

Question **4**

Correct

Marked out of 1.00

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 |

**Answer:** (penalty regime: 0 %)

|  |  |
| --- | --- |
| 1  2  3  4  5 ▼  6 ▼  7  8 ▼  9  10 ▼  11  12  13  14  15 | a=input() b=0  c=0 d=0  for char in a: if(char.isdigit()):  b+=1  elif(char.isalpha()): c+=1  else:  d+=1  print(c) print(b) print(d) |
|  | |

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

Passed all tests! 

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** | **Result** |
| Yn PYnative | True |

**Answer:** (penalty regime: 0 %)

|  |  |
| --- | --- |
| 1  2  3 ▼  4  5 ▼  6 | a=input() b=input()  if a in b or b in a: print("True")  else:  print("False") |
|  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | Yn PYnative | True | True |  |
|  | Ynf PYnative | False | False |  |

Passed all tests! 

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 |

**Answer:** (penalty regime: 0 %)

|  |  |
| --- | --- |
| 1  2  3 ▼  4  5 ▼  6  7 | a=input() s=a.split() if(len(s)>=2):  sw=s[1].upper() else:  sw="LESS"  print(sw) |
|  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | Wipro Technologies Bangalore | TECHNOLOGIES | TECHNOLOGIES |  |
|  | Hello World | WORLD | WORLD |  |
|  | Hello | LESS | LESS |  |

Passed all tests! 

Question **7**

Correct

Marked out of 1.00

Given a string S which is of the format [USERNAME@DOMAIN.EXTENSION,](mailto: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](mailto:abcd@gmail.com) Output:

com gmail abcd

### For example:

|  |  |
| --- | --- |
| **Input** | **Result** |
| [arvijayakumar@rajalakshmi.edu.in](mailto:arvijayakumar@rajalakshmi.edu.in) | edu.in rajalakshmi arvijayakumar |

**Answer:** (penalty regime: 0 %)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | a=input() u,e=a.split('@')  f,e=e.split('.',1) print(e)  print(f)  print(u) |
|  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | [abcd@gmail.com](mailto:abcd@gmail.com) | com gmail abcd | com gmail abcd |  |

Passed all tests! 

Question **8**

Correct

Marked out of 1.00

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:

1. <= 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.

**Answer:** (penalty regime: 0 %)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 ▼  7 ▼  8  9 ▼  10  11  12 | a=input() b=input() n=int(input()) cc=""  count=0  for char in a:  if char in b and char not in cc: cc+=char  if(len(cc)==n): break  print(cc) |
|  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | abcbde cdefghbb 3 | bcd | bcd |  |

Passed all tests! 

Question **9**

Correct

Marked out of 1.00

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

## Sample Input 1

a2b4c6

## Sample Output 1

aabbbbcccccc

**Answer:** (penalty regime: 0 %)

|  |  |
| --- | --- |
| 1 ▼ | def expand\_string(s): |
| 2 | result = '' |
| 3 | i = 0 |
| 4 ▼ | while i < len(s): |
| 5 | char = s[i] |
| 6 | i += 1 |
| 7 | num = '' |
| 8 ▼ | while i < len(s) and s[i].isdigit(): |
| 9 | num += s[i] |
| 10 | i += 1 |
| 11 | result += char \* int(num) |
| 12 | return result |
| 13 |  |
| 14 | s = input() |
| 15 | print(expand\_string(s)) |
| 16 |  |
|  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | **Expected** | **Got** |  |
|  | a2b4c6 | aabbbbcccccc | aabbbbcccccc |  |
|  | a12b3d4 | aaaaaaaaaaaabbbdddd | aaaaaaaaaaaabbbdddd |  |

Passed all tests! 

Question **10**

Correct

Marked out of 1.00

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 | 0 |
| aaaa | -1 |
| aaaA | 1 |
| abs |  |
| Abz |  |
| abcdefg |  |
| AbCdEfF |  |

**Answer:** (penalty regime: 0 %)

|  |  |
| --- | --- |
| 1  2 ▼  3  4  5 ▼  6  7 ▼  8  9 ▼  10 | t=int(input())  for i in range(t): str1=input().upper() str2=input().upper() if(str1<str2):  print("-1") elif(str1>str2):  print("1") else:  print("0") |
|  | |

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

Passed all tests! 

[◄ Week4\_Coding](http://118.185.187.137/moodle/mod/quiz/view.php?id=1064&forceview=1)

Jump to...

[Week6\_Coding ►](http://118.185.187.137/moodle/mod/quiz/view.php?id=1070&forceview=1)