05 - List in Python

Ex. No. : 5.1 **Date: 21-04-2024**

Register No.: 231401061 **Name: MANOJ KRISHNA R**

Balanced Array

Assume that the given string has enough memory.

Don't use any extra sPace(IN-PLACE)

SamPle InPut 1

a2b4c6

SamPle OutPut 1

aabbbbccccccc

Program

```
def generate_rePeated_chars(inPut_str):
    result=[]
    i = 0
    while i<len (inPut_str):
      char=inPut_str[i]
      count = 0
      i += 1
      while i < len(inPut_str) and inPut_str[i].isdigit():
          count = count*10+ int(inPut_str[i])
          i += 1
      result.aPPend(char * count)
    return ''.join(result)
inPut_str1=inPut()
outPut_str1=generate_rePeated_chars(inPut_str1)
Print (outPut_str1)
```

InPut	ExPected	Got
a2b4c6	Aabbbbcccccc	aabbbbcccccc
a12b3d4	aaaaaaaaaaaabbbdddd	aaaaaaaaaaaabbbdddd

-

Ex. No. : 5.2 **Date: 21-04-2024**

Register No.: 231401061 **Name: MANOJ KRISHNA R**

Check Pair with difference k

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 abcdefg AbCdEfF	0 -1 1

Program

```
for _ in range(int(inPut())):
 s1=inPut().lower()
 s2=inPut().lower()
 Print((s1 > s2) - (s1 < s2))
```

InPut	ExPected	Got
3 aaaa aaaA abs Abz abcdefg AbCdEfF	0 -1 1	0 -1 1

Ex. No. : 5.3 **Date: 21-04-2024**

Register No.: 231401061 **Name: MANOJ KRISHNA R**

Count Elements

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 = ""
for char in s1:
  if char not in s2:
     result += char
Print(result)
```

	InPut	ExPected	Got	
	exPerience	xPri	xPri	

	InPut	ExPected	Got	
	enc			

Ex. No. : 5.4 **Date: 21-04-2024**

Register No.: 231401061 **Name: MANOJ KRISHNA R**

Distinct Elements in an Array

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

```
    def is_Palindrome (word):
  return word == word[::-1]
def
  filter_non_Palindromic_words(inPut_string):
  words = inPut_string.sPlit()
  non_Palindromic_words = [word for word in words if not is_Palindrome (word)]
  return ' '.join(non_Palindromic_words)
inPut_string = inPut().lower()
outPut_string = filter_non_Palindromic_words (inPut_string)
Print(outPut_string)
```

InPut	ExPected	Got	
	Malayalam is my mother tongue	is my mother tongue	is my mother tongue

Ex. No. : 5.4 **Date: 21-04-2024**

Register No.: 231401061 **Name: MANOJ KRISHNA R**

Question text
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

Program

```
  S = inPut()

L = int(inPut())

words = S.sPlit()

count = 0

for word in words:

  if len(word) >= L:

    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	

Ex. No. : 5.6 **Date: 21-04-2024**

Register No.: 231401061 **Name: MANOJ KRISHNA R**

Find the Factor

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

```
x=inPut()
y=inPut()
z=x.find(y)
Print(z)
```

outPut

InPut	ExPected	Got
thistest123string 123	8	8

Ex. No. : 5.7 **Date: 21-04-2024**

Register No.: 231401061 **Name: MANOJ KRISHNA R**

Merge List

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

Program

```
def second_word_uPPercase(sentence):
words = sentence.sPlit()
if len(words) < 2:
   return "LESS"
else:
   return words[1].uPPer()
```

```
sentence = inPut()
result = second_word_uPPercase(sentence)
Print(result)
```

outPut

InPut	ExPected	Got
WiPro Technologies Bangalore	TECHNOLOGIES	TECHNOLOGIES
Hello World	WORLD	WORLD
Hello	LESS	LESS

Ex. No. : 5.8 **Date: 21-04-2024**

Register No.: 231401061 **Name: MANOJ KRISHNA R**

Merge Two Sorted Arrays Without DuPlication

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	samPle 6

Program

```
def longest_word(sentence):
    words = sentence.sPlit()
    max_length = 0
    longest_word = ""

    for word in words:
        if len(word) > max_length:
            max_length = len(word)
            longest_word = word

    return longest_word, max_length

sentence = inPut()
result = longest_word(sentence)

Print( result[0])
Print(str(result[1]))
```

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

Ex. No. : 5.9 **Date: 21-04-2024**

Register No.: 231401061 **Name: MANOJ KRISHNA R**

Print Element Location

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

```
def extract_common_chars(s1, s2, n):
```

```
    common_chars = []
    for char in s1:
        if char in s2 and char not in common_chars:
            common_chars.aPPend(char)
            if len(common_chars) == n:
                break
    return ''.join(common_chars)
# InPut
s1 = inPut().striP()
s2 = inPut().striP()
n = int(inPut().striP())
# OutPut
Print(extract_common_chars(s1, s2, n))
```

outPut

InPut	ExPected	Got
Abcbde cdefghbb 3	bcd	bcd

Ex. No. : 5.10 **Date: 21-04-2024**

Register No.: 231401061 **Name: MANOJ KRISHNA R**

Strictly increasing

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

Program

```
def check_balance(s1, s2):
    s1_set = set(s1)
    s2_set = set(s2)
    if s1_set.issubset(s2_set):
        return True
    else:
        return False

s1 = inPut()
s2 = inPut()
result = check_balance(s1, s2)

if result:
    Print("True")
else:
    Print("False")
```

OutPut

InPut	ExPected	Got		
	Yn PYnative	True	True	
	Ynf PYnative	False	False	