.



**05 - List in Python**

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

1

**Ex. No. : 5.1 Date:11/4/24**



**Register No: 231401037 Name:ILANKO M**

# 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

aabbbbcccccc

# 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)**

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

2

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

3



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



**Ex. No. : 5.2 Date:11/4/24**



**Register No: 231401037 Name:ILANKO M**

**Count all letters, digits, and special symbols**

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 |

# Program

**a=input()**

.

**l=sum(c.isalpha() for c in a)**

**d=sum(c.isdigit() for c in a)**

**s=sum(not c.isalnum() for c in a)**

**print(l)**

**print(d)**

**print(s)**

4

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

5



| **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 |  |

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

6



**Ex. No. : 5.3 Date:11/4/24**

**Register No: 231401037 Name:ILANKO M**

# Find key value or not

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:

Take 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

**For example:**

| **Input** | **Result** |
| --- | --- |
| break | break is a keyword |
| IF | IF is not a keyword |

PROGRAM

a=input()

if a in ["break","case","continue","default","defer","else","for","func","goto","if","map","range","return","struct","type","var"]:

print(a,"is a keyword")

else:

print(a,"is not a keyword")

| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | break | break is a keyword | break is a keyword |  |
|  | IF | IF is not a keyword | IF is not a keyword |  |

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

7



**Ex. No. : 5.4 Date:11/4/24**

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

8



**Register No: 231401037 Name: ILANKO M**

# 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:11/4/24**

**Register No: 231401037 Name:ILANKO M**

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

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

9

after Kenyattas inauguration police elsewhere capital, Nairobi, tried opposition holding peaceful

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

10



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:11/4/24**

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

11



**Register No: 231401037 Name:ILANKO M**

# 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:11/4/24**



**Register No: 231401037 Name:ILANKO M**

# 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()

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

12

sentence = input()

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

13



result = second\_word\_uppercase(sentence) print(result)

output



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

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

14



**Ex. No. : 5.8 Date:11/4/24**



**Register No: 231401037 Name:ILANKO M**

# 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]))

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

15

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

16



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **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:11/4/24**



**Register No: 231401037 Name:ILANKO M**

# Reverse a string without affecting secial character

**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:**

| **Input** | **Result** |
| --- | --- |
| A&x# | x&A# |

Program

def reverse\_string\_with\_special\_chars(s):

.

# Convert the string to a list for easy manipulation

s\_list = list(s)

# Initialize pointers

left = 0

right = len(s\_list) - 1

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

17

while left < right:



# Check if characters at pointers are alphabets

if s\_list[left].isalpha() and s\_list[right].isalpha():

# Swap alphabetic characters

s\_list[left], s\_list[right] = s\_list[right], s\_list[left]

# Move pointers

left += 1

right -= 1

elif not s\_list[left].isalpha():

# Move left pointer if it's not an alphabet

left += 1

elif not s\_list[right].isalpha():

# Move right pointer if it's not an alphabet

right -= 1

# Convert the list back to a string

reversed\_string = ''.join(s\_list)

return reversed\_string

# Test the function with example input

input\_string = input()

reversed\_string = reverse\_string\_with\_special\_chars(input\_string)

print(reversed\_string)

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

18



| **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | A&B | B&A | B&A |

**Ex. No. : 5.10 Date:11/4/24**



**Register No: 231401037 Name:ILANKO M**

# 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")

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

19

Output

.

**Department of Computer Science and Business systems** | **Rajalakshmi Engineering College**

20



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