

## **05 - Strings in Python**

**Ex. No. : 5.1**

**Date: 17.04.24**

**Register No.: 231901040**

**Name: Ramanitharan S**

## **String characters balance Test**

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".

Input	Result
Yn	True
PYnative	

**For example:**

### **Program:**

```
a=input()
b=input()
if a in b or b in a:
    print("True")
else:
    print("False")
```

**Ex. No. : 5.2**

**Date: 17.04.24**

**Register No.: 231901040**

**Name: Ramanitharan S**

---

## **Decompress the String**

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:**

```
s=input()
r=""
i=0
while i< len(s):
    char=s[i]
    i+=1
    num=""
    while i<len(s) and s[i].isdigit():
        num+=s[i]
        i+=1
    r+=char*int(num)
print(r)
```

**Ex. No. : 5.3**

**Date: 17.04.24**

**Register No.: 231901040**

**Name: Ramanitharan S**

### **First N Common Chars**

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:

$2 \leq N \leq 10$

$2 \leq \text{Length of S1, S2} \leq 1000$

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 i in a:
    if i in bset and i not in cc:
        cc.append(i)
        c=c+1
        if(c==n):
            break
s="".join(cc)
print(s)
```

**Ex. No. : 5.4**

**Date: 17.04.24**

**Register No.: 231901040**

**Name: Ramanitharan S**

## **Username Domain Extension**

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:**

vijayakumar.r@rajalakshmi.edu.in

### **Output:**

edu.in

rajalakshmi

vijayakumar.r

### **Program:**

```
s=input()
at=s.index('@')
dot=s.index('.')
username=s[:at]
domain=s[at+1:dot]
exten=s[dot+1:]
print(exten)
print(domain)
print(username)
```



**Ex. No. : 5.5**

**Date: 17.04.24**

**Register No.: 231901040**

**Name: Ramanitharan S**

## **Count Chars**

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

**For example:**

Input	Result
rec@12	3
3	3
	1

**Program:**

```
x=input()
a,b,c=0,0,0
for i in x:
    if(i.isalpha()):
        a+=1
    elif(i.isalnum()):
        b+=1
    else:
        c+=1
print(a,b,c,sep="\n")
```



**Ex. No. : 5.6**

**Date: 17.04.24**

**Register No.: 231901040**

**Name: Ramanitharan S**

---

### **Reverse String**

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:**

```
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)
```

**Ex. No. : 5.7**

**Date: 17.04.24**

**Register No.: 231901040**

**Name: Ramanitharan S**

## **Longest Word**

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:**

```
sen=input()
words=sen.split()
l=""
maxi=0
for word in words:
    if(len(word)>maxi):
        l=word
        maxi=len(word)
print(l,maxi,sep="\n")
```

**Ex. No. : 5.8**

**Date: 17.04.24**

**Register No.: 231901040**

**Name: Ramanitharan S**

## **Remove Palindrome Words**

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:**

```
s=input()
words=s.split()
x=""
for word in words:
    word=word.lower()
    if (word!=word[::-1]):
        print(word,end=" ")
```

**Ex. No. : 5.9**

**Date: 17.04.24**

**Register No.: 231901040**

**Name: Ramanitharan S**

## **Remove Characters**

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()
```

```
x="".join(char for char in s1 if char not in s2)
```

```
print(x)
```

## Unique Names

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:

**Input:**

first  
second  
first  
third  
second

then your program should display:

**Output:**

first  
second  
third

**Program:**

```
l=[]  
while(True):  
    a=input()  
    if a!=" ":  
        l.append(a)  
    else:  
        break  
l=dict.fromkeys(l)
```

```
for i in l:  
    print(i)
```