# 06 - Strings in Python

**Ex. No. : 6.1** **Date: 09.06.2024**

**Register No.: 230701108** **Name: HARSHAVARDHANAN V**

---

# Count Chars

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

```
n=input()
c1=0
c2=0
c3=0
for i in n:
    if(i in
"qwertyuiopassdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM"):
        c1=c1+1
    elif(i in "1234567890"):
        c2=c2+1
    else:
        c3=c3+1
print(c1,c2,c3,sep="\n")
```

Sample Input 1

a2b4c6

Sample Output 1

aabbbbcccccc

**Ex. No. : 6.2** **Date: 09.06.2024**

**Register No.: 230701108** **Name: HARSHAVARDHANAN V**

---

# Decompress the String

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

```
s = input()
e= ""
i = 0
while i < len(s):
 c= s[i]
 if c.isalpha():
 n= ""
 i += 1
 while i < len(s) and s[i].isdigit():
 n+= s[i]
 i += 1
 e+= c * int(n)
 else:
 e+= c
 i += 1
print(e)
```

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.

**Ex. No. : 6.3** **Date: 09.06.2024**

**Register No.: 230701108** **Name: HARSHAVARDHANAN V**

---

# 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.

```
S1 = input().strip()
S2 = input().strip()
N = int(input().strip())
c= []
p= set()
for r in S1:
 if r in S2 and r not in p:
 c.append(r)
 p.add(r)
 if len(c) == N:
 break
print(''.join(c))
```

Sample Input 1
experience
enc

Sample Output 1
xpri

**Ex. No. : 6.4** **Date: 09.06.2024**

**Register No.: 230701108** **Name: HARSHAVARDHANAN V**

---

# Remove Characters

Given two Strings s1 and s2, remove all the characters from s1 which is present in s2.

Constraints
1<= string length <= 200

```
n=input()
m=input()
k=""
for i in n:
 if(i not in m):
 k=k+i
print(k)
```

For example:

| Input | Expected |
| --- | --- |
| Malayalam is my mother tongue | is my mother tongue |
| He did a good deed | he good |

**Ex. No. : 6.5** **Date: 09.06.2024**

**Register No.: 230701108** **Name: HARSHAVARDHANAN V**

---

# 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

```
s1 =  input()
s= s1.lower()
li = s.split()
ans = []
def isPal(s):
    rev = s[::-1]
    if(rev==s):
        return False
    else:
        return True
for i in li:
    if(isPal(i)):
        ans.append(i)
for i in ans:
    print(i, end=" ")
```

For example:

Input  Result
Wipro Technologies Bangalore
TECHNOLOGIES
Hello World
WORLD
Hello
LESS

**Ex. No. : 6.6** **Date: 09.06.2024**

**Register No.: 230701108** **Name: HARSHAVARDHANAN V**

---

# Return Second World in Uppercase

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.

```
s = input()
a = s.split()
if len(a)>1:
 f=a[1].upper()
 print(f)
else:
 print("LESS"
```

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#

**Ex. No. : 6.7** **Date: 09.06.2024**

**Register No.: 230701108** **Name: HARSHAVARDHANAN V**

---

# 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.

```
s=input()

letters=[c for c in s if c.isalpha()]

letters.reverse()

result=""

for c in s:

 if c.isalpha():

 result+=letters.pop(0)

 else:

 result+=c

print(result)
```

For example:

| Input | Result |
| --- | --- |
| Yn | |
| PYnative | |
| True | |

**Ex. No. : 6.8** **Date: 09.06.2024**

**Register No.: 230701108** **Name: HARSHAVARDHANAN V**

---

# 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".

```
n=input()
x=input()
if(x.find(n) == -1):
    print("False")
else:
    print("True")
```

**Input:**
first
second
first
third
second

then your program should display:

**Output:**
first
second
third

**Ex. No. : 6.9** **Date: 09.06.2024**

**Register No.: 230701108** **Name: HARSHAVARDHANAN V**

---

# 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:

```
s=[]
for i in range(5):
a=input()
if a not in s:
s.append(a)
for i in s:
print(i)
```

Example Input/Output 1:

**Input**:

vijayakumar.r@rajalakshmi.edu.in

**Output**:

edu.in
rajalakshmi
vijayakumar.r

**Ex. No. : 6.10** **Date: 09.06.2024**

**Register No.: 230701108** **Name: HARSHAVARDHANAN V**

---

# 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

```
n=input()
s=""
d=""
w=""
t=n.find('.')
r=n.find("@")
print(n[t+1:],n[r+1:t],n[:r],sep='\n')
```