Ex. No. : 6.1 **Date:12/04/2024**

Register No.: 231901004 **Name: V AKASH DURAI**

Count Chars

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

Code:

```
string = input()
letters = 0
digits = 0
special_symbols = 0
for char in string:
  if char.isalpha():
    letters += 1
  elif char.isdigit():
    digits += 1
  else:
    special_symbols += 1
print(letters)
print(digits)
print(special_symbols)
```

Ex. No. : 6.2 Date:12/04/2024

Register No.: 231901004 Name: V AKASH DURAI

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

Code:

```
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. : 6.3 Date:12/04/2024

Register No.: 231901004 Name: V AKASH DURAI

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 <= 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.

Code:

```
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. : 6.4 Date:12/04/2024

Register No.: 231901004 Name: V AKASH DURAI

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

Code:

```
x=str(input())
y=str(input())
print(x.translate({ord(i):'' for i in y}))
```

Ex. No. : 6.5 Date: 12/04/2024

Register No.: 231901004 Name: V AKASH DURAI

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

Code:

```
s=str(input())
a=s.split(' ')
b=[]
for i in range(len(a)):
    a[i]=a[i].lower()
    if a[i][::1]!=a[i][::-1]:
        b.append(a[i])
for i in b:
    print(i,end=' ')
```

Ex. No. : **6.6** **Date:12/04/2024**

Register No.: **231901004** **Name: V AKASH DURAI**

Revers 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#

Code:

```
s = input().strip()
s = list(s)
left, right = 0, len(s) - 1
while left < right:
   if not s[left].isalpha():
      left += 1
   elif not s[right].isalpha():
      right -= 1
   else:
      s[left], s[right] = s[right], s[left]
      left += 1
      right -= 1
result = ''.join(s)
print(result)
```

Ex. No. : 6.7 Date:12/04/2024

Register No.: 231901004 Name: V AKASH DURAI

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".

For example:

Input	Result
Yn PYnative	True

Code:

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

Ex. No. : 6.8 Date:12/04/2024

Register No.: 231901004 Name: V AKASH DURAI

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
```

Code:

```
words = []
while True:
    word = input().strip()
    if word == "":
        break
    words.append(word)
seen_words = set()
for word in words:
    if word not in seen_words:
        print(word)
        seen_words.add(word)
```

Ex. No. : 6.9 **Date: 12/04/2024**

Register No.: 231901004 **Name: V AKASH DURAI**

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
```

Code:

```
s=str(input())
a=s.split('@')
if a[-1].count('.')==1:
    b=a[-1].split('.')
else:
    l=a[-1].split('.')
    b=[l[0],l[-2]+'.'+l[-1]]
u=a[0]
d=b[0]
e=b[-1]
print("%s\n%s\n%s"%(e,d,u))
```