WEEEK-05

1. **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().strip()**

**ch=list(s)**

**left,right=0,len(ch)-1**

**while left<right:**

**if not ch[left].isalpha():**

**left+=1**

**elif not ch[right].isalpha():**

**right-=1**

**else:**

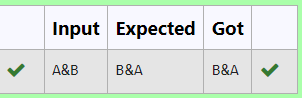
**ch[left],ch[right]=ch[right],ch[left]**

**left+=1**

**right-=1**

**print("".join(ch))**

**OUTPUT:**

****

**2.** 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().strip()**

**l=int(input())**

**w=s.split()**

**c=0**

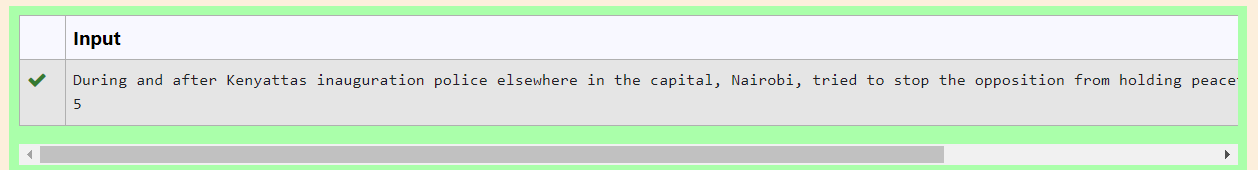
**for ws in w:**

**if len(ws)>=l:**

**c+=1**

**print(c)**

**OUTPUT:**

****

**3.** 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:**

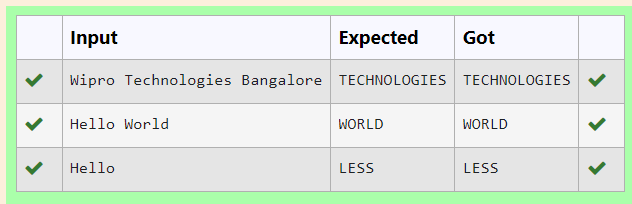
**return words[1].upper()**

**input\_sentence =str(input())**

**result= second\_word\_uppercase(input\_sentence)**

**print(result)**

**OUTPUT:**

****

**4.** 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:**

**s1=input().strip()**

**s2=input().strip()**

**n=int(input())**

**r=''**

**c=0**

**for char in s1:**

**if char in s2 and char not in r:**

**r+=char**

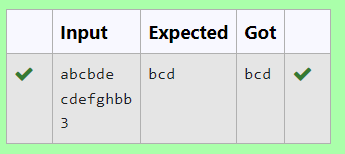
**c+=1**

**if c==n:**

**break**

**print(r)**

**OUTPUT:**

****

5. 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:**

**a=input()**

**a=a.split()**

**m=''**

**for i in a:**

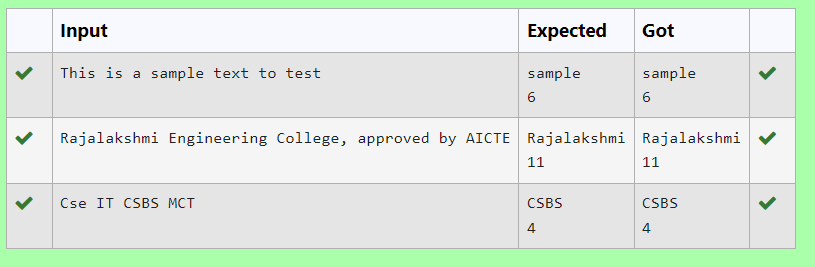
**if len(i)>len(m):**

**m=i**

**print(m)**

**print(len(m))**

**OUTPUT:**

****

6.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()**

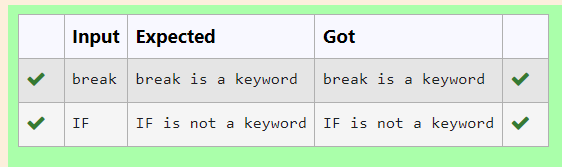
**if(a=="break" or a=="case" or a=="continue" or a=="default" or a=="defer" or a=="else" or a=="for" or a=="func" or a=="goto" or a=="map" or a=="if"or a=="range" or a=="return" or a=="struct" or a=="type" or a=="var"):**

**print(a,"is a keyword")**

**else:**

**print(a,"is not a keyword")**

**OUTPUT:**

****

**7.** 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:**

**a = list(input())**

**l = []**

**rep = []**

**i = 0**

**while i < len(a):**

**if a[i].isalpha():**

**l.append(i)**

**i += 1**

**else:**

**n = 0**

**while i < len(a) and not a[i].isalpha():**

**n = n \* 10 + int(a[i])**

**i += 1**

**rep.append(n)**

**s=''**

**j=0**

**for i in a:**

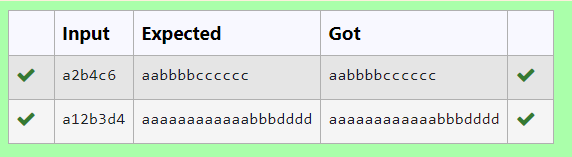
**if i.isalpha():**

**s+=i\*rep[j]**

**j+=1**

**print(s)**

**OUTPUT:**

****

**8.** 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:**

**a=input()**

**b=input()**

**# c=0**

**# for i in a:**

**# if(i.isalpha()):**

**# c+=1**

**# else:**

**# break**

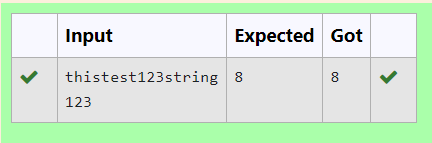
**if b in a:**

**print(a.index(b))**

**else:**

**print(-1)**

**OUTPUT:**

****

**9.** 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()**

**b=0**

**d=0**

**ss=0**

**for i in a:**

**if i.isalpha():**

**b+=1**

**elif i.isnumeric():**

**d+=1**

**else:**

**ss+=1**

**print(b)**

**print(d)**

**print(ss)**

**OUTPUT:**

****

**10.** 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 are\_strings\_balanced(s1, s2):**

**# Convert both strings to sets to remove duplicates**

**s1\_set = set(s1)**

**s2\_set = set(s2)**

**# Check if all characters in s1 are present in s2**

**return s1\_set.issubset(s2\_set)**

**# Example usage:**

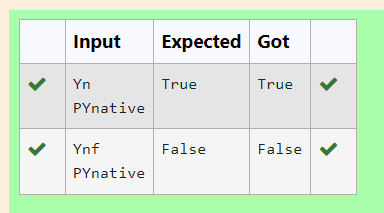
**s1 = input()**

**s2 = input()**

**result = are\_strings\_balanced(s1, s2)**

**print(result) # Output: True**

**OUTPUT:**

****