**WEEK 5-**

1.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:**

String1 = input()

String2 = input()

index = String1.find(String2)

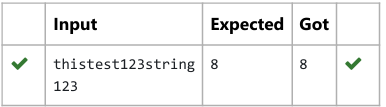
if index != -1:

print(index)

else:

print("-1")

**OUTPUT:**

****

2. 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

**PROGRAM:**

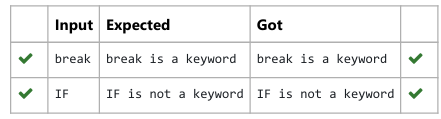
x=input() a=["break","case", "continue","default", "defer","else", "for" "map", "range", "return", "struct", "type", "var"]

if (x in a):

print(x,"is a keyword")   
else:

print(x,"is not a keyword")

**OUTPUT:**

****

3. Given a string S which is of the format USERNAME@DOMAIN.EXTENSION, the program must print the EXTENSION, DOMAIN, USERNAME in the reverse order.

Example Input/Output 1:

Input: [abcd@gmail.com](mailto:abcd@gmail.com)

Output:

Com

gmail

abcd

**PROGRAM:**

s=input()

at\_index=s.index('@')

dot\_index=s.index('.')

username=s[:at\_index]

domain=s[at\_index + 1:dot\_index]

extension=s[dot\_index+1:]

print(extension)

print(domain)

print(username)

**OUTPUT:**

****

4. 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:

During and after Kenyattas inauguration police elsewhere in the capital, Nairobi, tried to stop the opposition from holding peaceful demonstrations.

5

Output:

13

**PROGRAM:**

S = input()

L = int(input())

words = S.split() c

ount = 0

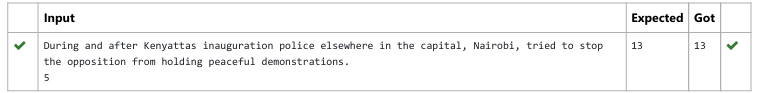
for word in words:

if len(word) >= L:

count += 1

print(count)

**OUTPUT:**

****

5. 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".

**PROGRAM:**

s1 = input()

s2 = input()

set\_s1 = set(s1)

set\_s2 = set(s2)

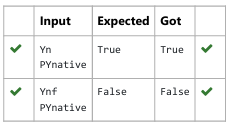
if set\_s1.issubset(set\_s2):

print("True")

else:

print("False")

**OUTPUT:**

****

6. 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.

Example Input/Output 1:

Input: abcbde

cdefghbb

3

Output:

bcd

**PROGRAM:**

a=input()

b=input()

n=int(input())

s=[]

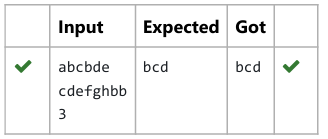
for i in a:

if i in b and i not in s:

s.append(i)

print("".join(s)[:n])

**OUTPUT:**

****

7. Robert  is having 2 strings consist of uppercase & lowercase english letters. Now he want to compare those two strings lexicographically. The letters' case does not matter, that is an uppercase letter is considered equivalent to the corresponding lowercase letter.

**PROGRAM:**

t=int(input())

for \_ in range(t):

str1=input().lower()

str2=input().lower()

if str1<str2:

print("-1")

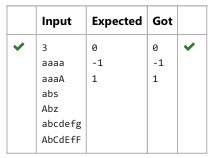
elif str1>str2:

print(“1”)

else:

print("0")

**OUTPUT:**

****

8. 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 expand\_string(s):

Result= ‘ ‘

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

result+=char\*int(num)

return result

s=input()

print(expand\_string(s))

**OUTPUT:**

****

9.  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

**PROGRAM:**

S=input()

S\_list=list(S)

start=0

end=len(S\_list)-1

While start<end:

if not S\_list[start].isalpha():

Start +=1

elif not S\_list[end].isalpha():

End-=1

else:

S\_list[start],S\_list[end]=S\_list[end],S\_list[start]

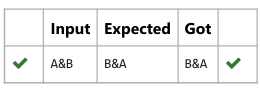
Start+=1

End -=1

Reversed\_string=’’.join(S\_list)

Print(Reversed\_string)

**OUTPUT:**

****

10.Write a program to count all the letters,digit and special symbols respectively from a given string

**PROGRAM :**

string = input()

letters = 0

digits = 0

special\_symbols = 0

for char in string:

if char.isalpha():

letters += 1

elif char.isdigit():

digits += 1

else:

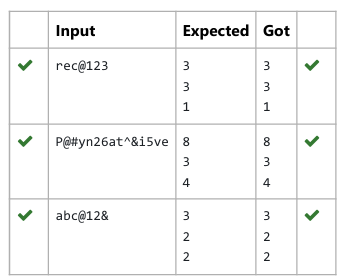
special\_symbols += 1

print(letters)

print(digits)

print(special\_symbols)

**OUTPUT:**

****