WEEK - 6

Strings

1) Find if a String2 is a substring of String1. If it is, return the index of the first occurrence. else return -1.

Test	Result
thistest123string 123	8

PROGRAM :

```
a=input()
b=input()
print(a.find(b)
```

2) Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.An input string is valid if Open brackets must be closed by the same type of brackets & Open brackets must be closed in the correct order.

s consists of parentheses only '()[]{}'.

Test	Result
print(ValidParenthesis("()"))	true
print(ValidParenthesis("()[]{}"))	true
print(ValidParenthesis("(]"))	false

PROGRAM :

```
def ValidParenthesis(s):
        a=[]
        b={') ' : '(', '}' : '{', ']' : '['}
        v=True
        for i in s:
                if i in b.values():
                        a.append(i)
```

```
            elif i in b.keys():
            if a and a[-1]= =b[i]:
                    a.pop()
            else:
                    v=False
                    break
    if v and a:
            v=False
    if v==True:
            return 'true'
    else:
            return 'false'
```

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

Test	Result
a2b4c6	aabbbbccccccc

PROGRAM :

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

4) 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.

Input	Result
break	break is a keyword
IF	IF is not a keyword

PROGRAM :

```
s=input()
if s in('break', 'case', 'continue', 'default', 'defer', 'else', 'for', 'func', 'goto','if','map','range'):
        print(s, "is a keyword")
else:
        print(s, 'is not a keyword')
```

5) Given a non-empty string s and an abbreviation abbr, return whether the string matches with the given abbreviation.

The string "word" contains only the following valid abbreviations: ["word", "1ord", "w1rd", "wo1d", "wor1", "2rd", "w2d", "wo2", "1o1d", "1or1", "w1r1", "1o2", "2r1", "3d", "w3", "4"]

Notice that only the above abbreviations are valid abbreviations of the string "word". Any other string is not a valid abbreviation of "word".

Note:Assume s contains only lowercase letters and abbr contains only lowercase letters and digits.

Test	Result
internationalization i12iz4n	true

apple a2e	false

PROGRAM :

```
s=input()
a=input()
 n=len(s)
m=len(a)
i,j=0,0
while i<n and j<m:
        if a[j].isdigit():
                num=0
                while j<m and a[j].isdigit(): num=num*10+int(a[j])
                        j+=1
                i+=num
        else:
                i+=1
                j+=1
if i==n and j==m:
        print('true')
else:
        print('false')
```

6) Write a Python program to get one string and reverse a string. The input string is given as an array of characters char[].You may assume all the characters consist of printable ascii characters.

Test	Result
hello	olleh
Hannah	hannaH

PROGRAM :

```
a=input()
print(a[::-1])
```

7) A pangram is a sentence where every letter of the English alphabet appears at least once.Given a string sentence containing only lowercase English letters, return true if sentence is a pangram, or false otherwise.

Test	Result
print(checkPangram('thequickbrownfoxjumpsoverthelazydog'))	true

PROGRAM :

```
import string
def checkPangram(s):
a=set(string.ascii_lowercase)
b=set(s.lower())
return 'true' if a.issubset(b) else 'false'
```

8) 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	Result
arvijayakumar@rajalakshmi.edu.in	edu.in rajalakshmi arvijayakumar

PROGRAM :

```
s=input()
```

```
u,d=s.split('@')
 dp=d.split('.',1)
 if len(dp)<=2:
        dn, e=dp
print(e)
 print(dn)
 print(u)
```

9) Given a string, determine if it is a palindrome, considering only alphanumeric characters and ignoring cases.

Note: For the purpose of this problem, we define empty string as valid palindrome.

Input	Result
A man, a plan, a canal: Panama	1
race a car	0

PROGRAM :

```
s=input()
s1 =''.join(i.lower() for i in s if i.isalnum() )
s2=s1[::-1]
if s1==52:
        print('1')
else:
        print('0')
```

10) The program must accept the N series of keystrokes as string values as the input. The character ^ represents undo action to clear the last entered keystroke. The program must print the string typed after applying the undo operations as the output. If there are no characters in the string then print -1 as the output.

Input	Result

3 Hey ^ goooo^^glee^ lucke^y ^charr^ms ora^^nge^^^^	Hey google luckycharms -1

PROGRAM :

```
n=int(input())
 for i in range(n):
        s=input()
         a=[]
        for j in s:
                if j=='^':
                        if a:
                                a.pop()
                else:
                        a.append(j)
        b=''.join(a)
        if b=='':
                print(-1)
        else:
                print(b)
```