Python programming-CSA0814

DAY 6

(9 aug 24)

**1**.**CHECKINGISOGRAM OR NOT**

sen=input()

css=''.join(e for e in sen.lower() if e.isalnum)

iso=len(css)==len(set(css))

print(iso)

**Output**

Isopleth

True

**2**.**CHECK THE GIVEN SENENCE IS PANGRAM OR NOT**

import string

sentence = input("Enter a sentence: ")

css = ''.join(e.lower() for e in sentence if e.isalpha())

as = set(string.ascii\_lowercase)

ss = set(css)

pangram = as.issubset(ss)

print(pangram)

OUTPUT:

Enter sentence: the quick brown fox jumps over the lazy dog

--->TRUE

**3**.**MISSING LETTERS TO MAKE THE SENTENCE AS PANGRAM**

import string

sentence = input("Enter a sentence: ")

css = ''.join(e.lower() for e in sentence if e.isalpha())

as = set(string.ascii\_lowercase)

ss = set(css)

missing = as - ss

if missing:

print( ', '.join(sorted(missing)))

else:

print("The sentence is already a pangram.")

OUTPUT:

Enter a sentence: the quick brown fox jumps over the lazy

---> d,s

**4**.**CHECKING TWO STRINGS ARE ANAGRAM TO EACH OTHER**

str1 = input("Enter the first string: ")

str2 = input("Enter the second string: ")

css1 = ''.join(str1.lower().split())

css2 = ''.join(str2.lower().split())

ana= sorted(css) == sorted(css)

if ana:

print("it is anagrams.')

else:

print("it is not anagrams.')

OUTPUT:

enter a string1:silent

enter a string2:listen

-->they are anagrams

**5.GROUP ANAGRAMS FROM LIST**

from collections import defaultdict

words = ["listen", "silent", "enlist", "hello", "drowl"]

ag = defaultdict(list)

for word in words:

ssorted = ''.join(sorted(word))

ag[ssorted].append(word)

print(list(ag))

OUTPUT:

listen,silent,enlist

**6.ENCODE A MESSAGE BY SHIFTING A POSITION WITH ALPHABET**

message = "Hello World"

shift = 3

encoded\_message = ""

for char in message:

if char.isalpha():

start = ord('A') if char.isupper() else ord('a')

shifted = (ord(char) - start + shift) % 26

encoded\_char = chr(start + shifted)

encoded\_message += encoded\_char

else:

encoded\_message += char

print(f"Original message: {message}")

print(f"Encoded message: {encoded\_message}")

OUTPUT:

Original message: Hello World

Encoded message: Khoor Zruog

**7.PROGRAM TO GET PALINDROMIC SUBSTRING FROM GIVEN STRING**

string = "noonlight"

length = len(string)

result = []

for i in range(length):

for j in range(i + 1, length + 1):

substring = string[i:j]

if substring == substring[::-1]:

result.append(substring)

print("Palindromic substrings:", result)

OUTPUT:

Palindromic substrings: ['n', 'o', 'o', 'n', 'l', 'i', 'g','h','t', 'noon']

**8.MERGE TWO SORTED LIST**

list1 = [3, 1, 4, 2]

list2 = [5, 9, 7, 6]

merged\_list = list1 + list2

sorted\_list = sorted(merged\_list)

print( sorted\_list)

OUTPUT:

[1, 2, 3, 4, 5, 6,7, 9]

**9.ROTATE ELEMENTS IN LIST K TIMES**

lst = [1, 2, 3, 4, 5]

k = int(input("Enter the number of positions to rotate: "))

k = k % len(lst)

rotated\_list = lst[-k:] + lst[:-k]

print("Original list:", lst)

print("Rotated list:", rotated\_list)

OUTPUT:

Enter the number of positions to rotate: 2

Original list: [1, 2, 3, 4, 5]

Rotated list: [4, 5, 1, 2, 3]

**10.PRINT PAIR ELEMENTS WHICH ADD UPTO TARGET ELEMENT**

lst = [2, 4, 3, 7, 5, 1]

target = 8

pairs = []

for i in range(len(lst)):

for j in range(i + 1, len(lst)):

if lst[i] + lst[j] == target:

pairs.append((lst[i], lst[j]))

print("Pairs that sum up to",target,"is",pairs)

OUTPUT:

Pairs that sum up to 8 is [(3, 5), (7, 1)]