**Week7:TUPLES,SET AND ITS OPERATIONS**

1. The DNA sequence is composed of a series of nucleotides abbreviated as 'A', 'C', 'G', and 'T'.

For example, "ACGAATTCCG" is a DNA sequence.

When studying DNA, it is useful to identify repeated sequences within the DNA.

Given a string s that represents a DNA sequence, return all the 10-letter-long sequences (substrings) that occur more than once in a DNA

molecule. You may return the answer in any order.

**Example 1:**

Input: s = "AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT"

Output: ["AAAAACCCCC","CCCCCAAAAA"]

**Example 2:**

Input: s = "AAAAAAAAAAAAA"

Output: ["AAAAAAAAAA"]

**PROGRAM:**

def findRepeatedSequences(s):

sequences = {}

result = []

for i in range(len(s) - 9):

seq = s[i:i+10]

sequences[seq] = sequences.get(seq, 0) + 1

if sequences[seq] == 2:

result.append(seq)

return result

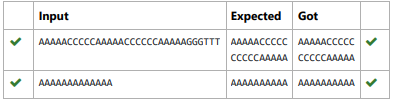
# Example usage

s1 = input()

for i in findRepeatedSequences(s1):

print(i)

**OUTPUT:**

****

2. Given a tuple and a positive integer k, the task is to find the count of distinct pairs in the tuple whose sum is equal to K.

**Sample Input :** t = (5, 6, 5, 7, 7, 8 ), K = 13

**Sample Output**: 2  
  
**PROGRAM:**

t = input().split(',')

k = int(input())

s = []

s1=[]

for i in range(len(t)-1):

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

if (int(t[i])+int(t[j]))==k and (t[i],t[j]) not in s1

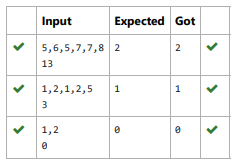
s1.append((t[i],t[j]))

s1.append((t[j],t[i]))

s.append((t[i],t[j]))

print(len(set(s)))

**OUTPUT:**

****

3. Given an array of strings words, return the words that can be typed using letters of the alphabet on only one row of American keyboard

In the American keyboard:

the first row consists of the characters "qwertyuiop",

the second row consists of the characters "asdfghjkl", and

the third row consists of the characters "zxcvbnm".

**Example 1:**

**Input:** words = ["Hello","Alaska","Dad","Peace"]

**Output:** ["Alaska","Dad"]

**Example 2:**

**Input:** words = ["omk"]

**Output:** []

**Example 3:**

**Input:** words = ["adsdf","sfd"]

**Output:** ["adsdf","sfd"]

**PROGRAM:**

def findWords(words):

row1 = set('qwertyuiop')

row2 = set('asdfghjkl')

row3 = set('zxcvbnm')

result = []

for word in words:

w = set(word.lower())

if w.issubset(row1) or w.issubset(row2) or w.issubset

result.append(word)

if len(result) == 0:

print("No words")

else:

for i in result:

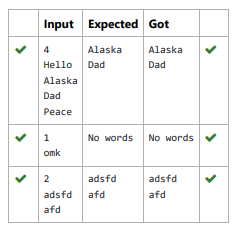
print(i)

a = int(input())

arr = [input() for i in range(a)]

findWords(arr)

**OUTPUT:**

****

4. Coders here is a simple task for you, Given string str. Your task is to check whether it is a binary string or not by using python set.

**Examples:**

**Input:** str = "01010101010"

**Output:** Yes

**Input:** str = "REC101"

**Output:** No

**PROGRAM:**

a = input()

try:

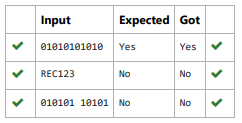
c = int(a)

print("Yes")

except:

print("No")

**OUTPUT:**

****

5. Given an array of integers nums containing n + 1 integers where each integer is in the range [1, n] inclusive.There is only one repeated number in nums, return this repeated number. Solve the problem using set.

**Example 1:**

**Input:** nums = [1,3,4,2,2]

**Output:** 2

**Example 2:**

**Input:** nums = [3,1,3,4,2]

**Output:** 3

**PROGRAM:**

n =input().split(" ")

n = list(n)

for i in range(len(n)):

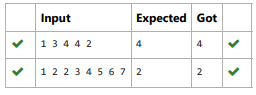
for j in range(i+1,len(n)):

if n[i] == n[j]:

print(n[i])

exit(0)

**OUTPUT:**

****