7/8/2021

**Rohan Gupta**

**PS No. 99005019**

**LTTS Intern**

**Python Assignment**

**Ques1. In a given list of elements, all elements are equal except the one. Find unique elements from list.**

myList = [10, 10, 10, 10, 12, 10]  
uniqueList = []  
for i in myList:  
 if i not in uniqueList:  
 uniqueList.append(i)  
for i in uniqueList:  
 print(i)

**Ques2. In a given** **list of elements, find the elements which is close to its mean**

def findClosest(arr, n, target):  
 # Corner cases  
 if target <= arr[0]:  
 return arr[0]  
 if target >= arr[n - 1]:  
 return arr[n - 1]  
 # Doing binary search  
 i = 0  
 j = n  
 mid = 0  
 while i < j:  
 mid = (i + j) // 2  
 if arr[mid] == target:  
 return arr[mid]  
 # If target is less than array  
 # element, then search in left  
 if target < arr[mid]:  
 # If target is greater than previous  
 # to mid, return closest of two  
 if mid > 0 and target > arr[mid - 1]:  
 return getClosest(arr[mid - 1], arr[mid], target)  
 # Repeat for left half  
 j = mid  
 # If target is greater than mid  
 else:  
 if mid < n - 1 and target < arr[mid + 1]:  
 return getClosest(arr[mid], arr[mid + 1], target)  
 i = mid + 1  
 # Only single element left after search  
 return arr[mid]  
  
  
def getClosest(val1, val2, target):  
 if target - val1 >= val2 - target:  
 return val2  
 else:  
 return val1  
  
  
# Driver code  
arr = [2, 5, 6, 7, 8, 8, 6]  
n = len(arr)  
sum\_elements = 0  
mean = 0  
for i in range(0, n):  
 sum\_elements = sum\_elements + arr[i]  
mean = sum\_elements / n  
target = mean  
print(findClosest(arr, n, target))

**Ques3. Find the average speed of vehicle, given the distance travelled for fixed time intervals**

arr = [0, 0.1, 0.5, 0.55, 0.35, 0.45, 0.90]  
n = len(arr)  
sum\_speed = 0  
mean = 0  
time = 1  
for i in range(0, n):  
 sum\_speed = sum\_speed + (arr[i]/time)  
 time = time + 1  
mean = sum\_speed / n  
print(mean)

**Ques4. Find the no.of people in a bus, given the data of people onboarding & alighting at each station**

onboarded = [10, 2, 3, 4, 5, 1, 2]  
alighted = [0, 3, 2, 1, 3, 2, 1]  
no\_of\_passenger = 0  
for i in range(0, len(onboarded)):  
 print("No. of passenger at station", (i+1), " is")  
 no\_of\_passenger = no\_of\_passenger + (onboarded[i] - alighted[i])  
 print(no\_of\_passenger)

**Ques5. Find the missing number, given the original list and modified one**

original\_list = [1, 2, 3, 4, 5, 6, 7, 8]  
modified\_list = [1, 2, 3, 6, 4, 7]  
for i in range(0,len(original\_list)):  
 if original\_list[i] not in modified\_list:  
 print(original\_list[i])

**Ques6. Find the difference between two lowest numbers in the list**

import sys  
  
  
def print\_2\_smallest(arr):  
 arr\_size = len(arr)  
 if arr\_size < 2:  
 print("Invalid Input")  
 return  
 first = second = sys.maxsize  
 for i in range(0, arr\_size):  
 # If current element is smaller than first then  
 # update both first and second  
 if arr[i] < first:  
 second = first  
 first = arr[i]  
 # If arr[i] is in between first and second then  
 # update second  
 elif arr[i] < second and arr[i] != first:  
 second = arr[i]  
 print("The difference between two smallest number is ", second-first)  
  
  
# Driver function to test above function  
arr = [14, 15, 1, 10, 25, 5]  
print\_2\_smallest(arr)  
arr = [14, 15, 1, 10, 25, 1]  
print\_2\_smallest(arr)

**Ques7. In a given list, count number of elements smaller than their mean.**

arr = [0, 0.1, 0.5, 0.55, 0.35, 0.45, 0.90]  
n = len(arr)  
sum\_elements = 0  
mean = 0  
for i in range(0, n):  
 sum\_elements = sum\_elements + arr[i]  
mean = sum\_elements / n  
print(mean)  
count = 0  
for i in range(0, len(arr)):  
 if arr[i] < mean:  
 print("Number smaller than mean is ", arr[i])  
 count = count+1  
print("Count of elements smaller then their mean is ", count)

1. **Correct the malformed time string, for e.g "5:70:65" to "6:11:05"**

def time\_correct\_format(hrs, minute, sec):  
 if int(sec) > 60:  
 sec = int(sec) - 60  
 minute = int(minute) + 1  
 if sec < 10:  
 sec = str(0) + str(sec)  
 sec = str(sec)  
 if int(minute) > 60:  
 minute = int(minute) - 60  
 hrs = int(hrs) + 1  
 if minute < 10:  
 minute = str(0) + str(minute)  
 minute = str(minute)  
 hrs = str(hrs)  
 print(hrs + ":" + minute + ":" + sec)  
  
  
time = "5:80:55"  
hr = time[0]  
minutes = time[2:4]  
second = time[5:7]  
time\_correct\_format(hr, minutes, second)  
time = "5:70:65"  
hr = time[0]  
minutes = time[2:4]  
second = time[5:7]  
time\_correct\_format(hr, minutes, second)

**2. Correct the malformed date string, for e.g. "45/8/2018" to "14/9/2018"**

# Check for Leap Year  
def check\_year(year):  
 if (year % 4) == 0:  
 if (year % 100) == 0:  
 if (year % 400) == 0:  
 return True  
 else:  
 return False  
 else:  
 return True  
 else:  
 return False  
  
  
no\_of\_days = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]  
date = "45/2/2016"  
day, month, year = date.split("/")  
flag = check\_year(int(year))  
if flag:  
 no\_of\_days[1] = no\_of\_days[1] + 1  
print(day, month, year)  
if int(day) > no\_of\_days[int(month) - 1]:  
 day = int(day) - no\_of\_days[int(month) - 1]  
 month = int(month) + 1  
 if month > 12:  
 year = int(year) + 1  
print(str(day) + "/" + str(month) + "/" + str(year))

**3. Convert IP address from "a.b.c.d" format into integer and vice versa**

def convert\_ip\_to\_integer(input\_string\_ip\_address):  
 ip\_list = list(input\_string\_ip\_address.split("."))  
 result = 0  
 j = 3  
 for i in ip\_list:  
 result = result + int(i)\*(256\*\*j)  
 j = j-1  
 print(result)  
  
  
def convert\_integer\_to\_ip(ip):  
 ip\_list = []  
 ip\_address = ""  
 ip\_list.append(ip & 0xFF)  
 ip\_list.append((ip >> 8) & 0xFF)  
 ip\_list.append((ip >> 16) & 0xFF)  
 ip\_list.append((ip >> 24) & 0xFF)  
 ip\_list = ip\_list[::-1]  
 for i in ip\_list:  
 ip\_address = ip\_address + str(i) + ":"  
 ip\_address = ip\_address[0:len(ip\_address)-1]  
 print(ip\_address)  
  
  
convert\_ip\_to\_integer("49.36.221.122")  
convert\_ip\_to\_integer("56.66.255.110")  
convert\_integer\_to\_ip(824499578)  
convert\_integer\_to\_ip(943914862)

**4. Check whether given string is isogram or not**

# Python program to check  
# if a word is isogram or not  
def check\_for\_isogram(word):  
 # Convert the word or sentence in lower case letters.  
 clean\_word = word.lower()  
  
 # Make an empty list to append unique letters  
 letter\_list = []  
  
 for letter in clean\_word:  
  
 # If letter is an alphabet then only check  
 if letter.isalpha():  
 if letter in letter\_list:  
 return False  
 letter\_list.append(letter)  
  
 return True  
  
  
print(check\_for\_isogram("Machine"))  
print(check\_for\_isogram("isogram"))  
print(check\_for\_isogram("Genesis"))  
print(check\_for\_isogram("Alphabet "))

**5. Given a string, find the Mexican wave**

def mexican\_wave(string):  
 result\_list = []  
 for i in range(len(string)):  
 string = string.lower()  
 temp\_list = list(string)  
 if temp\_list[i].isalpha():  
 temp\_list[i] = string[i].upper()  
 string = ''.join(temp\_list)  
 result\_list.append(string)  
 return result\_list  
  
  
input\_string = "genesis"  
final\_result = mexican\_wave(input\_string)  
print(final\_result)  
input\_string = "python"  
final\_result = mexican\_wave(input\_string)  
print(final\_result)

**6. Given a number, find the largest number by deleting single digit**

def max\_number(n, k):  
 for i in range(0, k):  
 ans = 0  
 i = 1  
 while n // i > 0:  
 temp = (n // (i \* 10)) \* i + (n % i)  
 i \*= 10  
 if temp > ans:  
 ans = temp  
 n = ans  
 return ans  
  
  
n = 657  
k = 1  
print(max\_number(n, k))  
n = 542  
k = 1  
print(max\_number(n, k))

**7. Given a number, find the largest number by shuffling the digits**

def print\_maximum\_shuffling(num):  
 counter = [0 for x in range(10)]  
 string = str(num)  
 for i in range(len(string)):  
 counter[int(string[i])] = counter[int(string[i])] + 1  
 result = 0  
 multiply = 1  
 for i in range(10):  
 while counter[i] > 0:  
 result = result + (i \* multiply)  
 counter[i] = counter[i] - 1  
 multiply = multiply \* 10  
 return result  
  
  
number = 38293367  
print(print\_maximum\_shuffling(number))  
number = 412567845  
print(print\_maximum\_shuffling(number))

**8. Compute the word frequency in given message**

s = "the lazy fox jump over the wall"  
string\_list = list(s.split(" "))  
temp\_l = []  
for i in string\_list:  
 if i not in temp\_l:  
 print("The frequency of word ", i, "is ", s.count(i))  
 temp\_l.append(i)

**9. RGB to Hex conversion and vice versa, e.g. (255,0,255) into 0xFF00FF**

def rgb\_to\_hex\_conversion(rgb):  
 return '%02x%02x%02x' % rgb  
  
  
def hex\_to\_rgb\_conversion(value):  
 value = value.lstrip('#')  
 lv = len(value)  
 return tuple(int(value[i:i + lv // 3], 16) for i in range(0, lv, lv // 3))  
  
  
print(hex\_to\_rgb\_conversion("FF65BA"))  
print(rgb\_to\_hex\_conversion((255, 0, 255)))

**10. Generate accumulated strings, e.g., abcd ==> A-Bb-Ccc-Dddd**

def accumulated\_string(input\_string):  
 temp\_string = ""  
 for i in range(0, len(input\_string)):  
 temp\_string = temp\_string + input\_string[i].upper()  
 for j in range(0, i):  
 temp\_string = temp\_string + input\_string[i]  
 if i != len(input\_string) - 1:  
 temp\_string = temp\_string + "-"  
 print(temp\_string)  
  
  
input\_string = "rohan"  
accumulated\_string(input\_string)  
input\_string = "genesis"  
accumulated\_string(input\_string)