**Assignment - 1**

1. In a given list of elements, all elements are equal except the one. Write a code to find the odd man out (Stray number)

mylist = [11,22,11,11]  
#mylist = ["abc","efg","abc","abc"] //lis  
  
  
temp = mylist[0]  
count = 1  
count1 = 0  
temp1 = mylist[1]  
  
for i in range(len(mylist)-1) :  
 if temp == mylist[i+1] :  
 count = count + 1  
 else :  
 temp1 = mylist[i+1]  
 count1 = count1 + 1  
if count < count1 :  
 print("The odd one in the list is",temp)  
 #print("The odd one in the list is " + temp)  
else :  
 print("The odd one in the list is ",temp1)  
 #print("The odd one in the list is " + temp1)

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

mylist = [21,1,2,3,4,5,6,7,8.5,9,10]  
sum = 0  
for i in range(len(mylist)) :  
 sum = sum + mylist[i]  
  
mean = sum/len(mylist)  
print("mean ", mean)  
  
mylist.sort()  
temp = mean - mylist[0]  
  
arr = []  
for i in range(len(mylist)):  
 arr.append(mean - mylist[i])  
  
temp = arr[0]  
for i in range(len(arr)) :  
 arr[i] = abs(arr[i])  
  
for i in range(len(arr)-1) :  
 if temp < arr[i+1] :  
 pass  
 else :  
 temp = arr[i+1]  
  
temp1 = arr.index(temp)  
print("number near to mean in list is ", mylist[temp1])

3. Find the average speed of vehicle, given the distance travelled for fixed time intervals, e.g.

[0, 0.1,0.25, 0.45, 0.55, 0.7, 0.9, 1.0]

distance = [0, 0.1, 0.25, 0.65, 1.25, 2.75, 5] #in terms of km  
time = 2 #hours  
avg\_speed = []  
  
for i in range(len(distance)) :  
 avg\_speed.append(distance[i]/time)  
  
print("avg speeds are -> ",avg\_speed)

4. Find the number of people in a bus, given the data of people onboarding & alighting at each station

n = int(input("Enter the number of stations :"))  
arr = []  
  
for i in range(n):  
 print("Please enter data for station ",i)  
 arr.append(int(input("Eneter the no. of people onboarding ")))  
 arr.append(int(input("Eneter the no. of people leaving ")))  
  
sum = 0  
for i in range(len(arr)):  
 if i%2 == 0:  
 sum = sum + arr[i]  
 else :  
 sum = sum - arr[i]  
  
print(sum)

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

mylist = [1,2,3,4,5,6,7,8,9]  
mylist1 = [1,2,3,4,6,7,8,9]  
  
mylist.sort()  
mylist1.sort()  
  
print(mylist)  
print(mylist1)  
mylist1.append(0)  
  
for i in range(len(mylist)) :  
 if mylist[i] == mylist1[i] :  
 pass  
 else :  
 print(mylist[i])  
 break

6. Find the difference between two lowest numbers in the list

mylist = [5,4,3,9,7]  
mylist1 = [11,133,5,9,16]  
  
temp = min(mylist)  
temp1 = min(mylist1)  
  
print(abs(temp-temp1))

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

mylist = [5,4,3,9,7,11,55]  
  
mylist.sort()  
sum = 0  
for i in range(len(mylist)) :  
 sum = sum + mylist[i]  
  
mean = sum/len(mylist)  
  
print("list = ",mylist)  
print("mean = ",mean)  
  
count = 0  
for i in range(len(mylist)) :  
 if mylist[i] < mean :  
 count = count + 1  
print(count)

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

date\_list = ["45/8/2018"]  
  
for i in date\_list:  
 d, m, y = i.split('/')  
  
d = int(d)  
m = int(m)  
y = int(y)  
#print(d,m,y)  
  
date = d % 31  
da = d // 31  
  
yr = 0  
month = m + da  
if month > 12 :  
 month = month % 12  
 yr = 1  
  
print(date,end="/")  
print(month,end="/")  
print(y+yr)

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

time\_list = ["5:70:65"]  
  
for i in time\_list:  
 h, m, s = i.split(':')  
  
h = int(h)  
m = int(m)  
s = int(s)  
  
print(h,m,s)  
secq = s / 60  
sec = s % 60  
  
min = secq + m  
minq = min / 60  
min = min % 60  
  
hr = int(minq) + h  
  
#print(sec , int(min), hr)  
print(hr,":",int(min),":",sec)

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

ip\_add = ["255.254.253.252"]  
print(type(ip\_add[0]), ip\_add)  
for i in ip\_add:  
 a, b, c, d = i.split('.')  
  
ip\_add = []  
ip\_add.append(int(a))  
ip\_add.append(int(b))  
ip\_add.append(int(c))  
ip\_add.append(int(d))  
  
#print(ip\_add)  
  
print(type(ip\_add[0]),ip\_add)  
  
ip\_addr = []  
  
ip\_addr.append(str(ip\_add[0]))  
ip\_addr.append(str(ip\_add[1]))  
ip\_addr.append(str(ip\_add[2]))  
ip\_addr.append(str(ip\_add[3]))  
  
ip = ip\_addr[0] +"."+ ip\_addr[1] +"."+ip\_addr[0] +"."+ ip\_addr[1]  
print(type(ip),ip)

4. Check whether given string is isogram or not

word="geek"  
list1=[]  
list1[:0]=word  
iso=[]  
for char in list1:  
 if char not in iso:  
 iso.append(char)  
if iso==list1:  
 print("isogram")  
else:  
 print("no")

5. Given a string, find the mexican wave

s='string'  
new=[]  
for i, val in enumerate(s[:]):  
 up=s[i].upper()  
 c=s[:i] + up + s[i+1:]  
 new.append(c)  
print(new)

6. Given a number, find the largest number by deleting single digit (order of digits will remain same)

number = 984768  
  
number = str(number)  
minimum\_num = number[0]  
position\_num = 0  
largest\_num = ""  
for i in range(len(number)-1):  
 if number[i+1] < minimum\_num:  
 minimum\_num = number[i+1]  
 position\_num = i+1  
for i in range(len(number)):  
 if position\_num == i:  
 continue  
 largest\_num += number[i]  
  
print("The largest number possible after deleting a single digit :", largest\_num)

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

number = 984768  
  
number1 = str(number)  
number2 = list(number1)  
largest = ""  
for i in range(len(number2) - 1):  
 for j in range(len(number2) - 1):  
 if number2[j] < number2[j + 1]:  
 temp = number2[j]  
 number2[j] = number2[j + 1]  
 number2[j + 1] = temp  
  
for i in range(len(number2)):  
 largest += number2[i]  
print("The largest number after shuffling the digits :", largest)

8. Compute the word frequency in given message

String1 = "The frequency is the number of occurrence"  
  
word\_list = String1.split(" ")  
frequency\_list = []  
i = 0  
for word in word\_list:  
 word = word.lower()  
 count = 0  
 for word1 in word\_list:  
 word1 = word1.lower()  
 if word == word1:  
 count += 1  
 frequency\_list.append(count)  
 i += 1  
  
print("the word list and the associated frequency of the word")  
print(word\_list)  
print(frequency\_list)

9. RGB to Hex conversion and vice versa, e.g. (255,0,255) into OxFF00FF

rgb = (255, 0, 255)  
rgb\_hex = '%02x%02x%02x' % rgb  
print("Hex value of the given RGB pair :", rgb\_hex)

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

String1 = "abcd"  
String2 = ""  
temp\_string = ""  
for i in range(len(String1)):  
 c = String1[i].upper()  
 temp\_string = c + String1[i] \* i  
 String2 += (temp\_string)  
 if i < len(String1) - 1 :  
 String2 += "-"  
print("Accumulated String :", String2)