NAME: PRACHI TANNA PS NO. 99005011

Contents

[SET 1 2](#_Toc76834139)

[1. Given a list of elements all elements are equal except one. Write a code to find the odd man out(Stray number) 2](#_Toc76834140)

[2. In the given list of elements, find the element which is close to its mean. 3](#_Toc76834141)

[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] 3](#_Toc76834142)

[4. Find the number of people in a bus, given the data of people onboarding and alighting at each station. 4](#_Toc76834143)

[5. Find the missing number given the original and modified one. 4](#_Toc76834144)

[6. Find the difference between two lowest numbers in the list. 4](#_Toc76834145)

[7. In a given list, count number of elements smaller than mean 5](#_Toc76834146)

[SET 2 6](#_Toc76834147)

[1. Correct the malformed time string, for e.g. “5:70:65” to “6:11:05” 6](#_Toc76834148)

[2. Correct the malformed date string, for e.g. “45/8/2018” to “14/9/2018” 6](#_Toc76834149)

[3. Convert IP address to integer and vice versa 7](#_Toc76834150)

[4. Check whether the given string is isogram or not. 8](#_Toc76834151)

[5. Given a string, find the Mexican wave. 8](#_Toc76834152)

[6. Given a number, find the largest number by deleting a single digit (Order will remain same) 9](#_Toc76834153)

[7. Given a number, find the largest number by shuffling it. 10](#_Toc76834154)

[8. Compute word frequency in given message 10](#_Toc76834155)

[9. Convert RGB to Hex and vice versa 11](#_Toc76834156)

[10. Generate accumulated strings, e.g. abcd == > A-Bb-Ccc-Dddd 11](#_Toc76834157)

# 

# SET 1

### Given a list of elements all elements are equal except one. Write a code to find the odd man out(Stray number)

mylist = [4, 2, 2, 2, 2]

a = mylist[0]

k = mylist[1]

if a != k:

p = mylist.count(a)

q = mylist.count(k)

if p > q:

print("The odd element is", k)

else:

print("The odd element is", a)

else:

for i in range(1, len(mylist)):

if a == mylist[i]:

e = mylist[i]

else:

f = mylist[i]

m = i + 1

b = mylist[m]

c = mylist.count(f)

d = mylist.count(b)

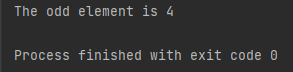
if c > d:

print("The odd element is", mylist[m])

else:

print("The odd element is", mylist[i])

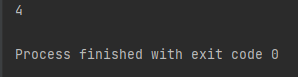
#### Output:



### In the given list of elements, find the element which is close to its mean.

list\_1 = [1, 2, 3, 4, 5, 6, 7]  
t = 0  
new\_list = []  
for i in list\_1:  
 t += i  
mean = t / len(list\_1)  
# print(mean)  
for i in range(0, len(list\_1)):  
 new\_list.append(mean - list\_1[i])  
 if new\_list[i] < 0:  
 new\_list[i] = new\_list[i] \* (-1)  
min = new\_list[0]  
for i in range(0, len(new\_list)):  
 if min > new\_list[i]:  
 min = new\_list[i]  
for i in range(0, len(new\_list)):  
 if min == new\_list[i]:  
 print(list\_1[i])  
 break

#### Output:



### 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, 0.45, 0.55, 0.7, 0.9, 1.0]

time = 1

total = 0

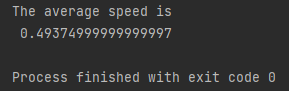
for i in distance:

total += i

avg\_speed = total/len(distance)

print("The average speed is \n", avg\_speed)

#### Output:



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

pas\_onboard = [30, 23, 35, 14, 45, 26, 19]

pas\_alight = [33, 25, 23, 44, 21, 16, 11]

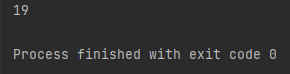
pas = 0

for x in range(len(pas\_onboard)):

pas += pas\_onboard[x] - pas\_alight[x]

print(pas)

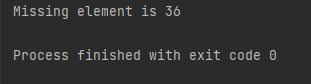
#### Output:



### Find the missing number given the original and modified one.

list\_p = [12, 23, 14, 36, 27, 18, 20]  
mod\_list = [12, 23, 14, 27, 18, 20]  
for i in range(0, len(list\_p)):  
 if list\_p[i] in mod\_list:  
 continue  
 else:  
 print("Missing element is", list\_p[i])

#### Output:



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

my\_list = [4, 8, 9, 3, 6, 1]

new\_list = []

while my\_list:

mini = my\_list[0]

for x in my\_list:

if x < mini:

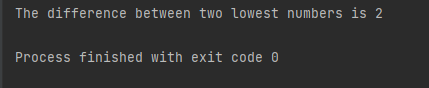
mini = x

new\_list.append(mini)

my\_list.remove(mini)

print("The difference between two lowest numbers is", abs(new\_list[1] - new\_list[0]))

#### Output:



### In a given list, count number of elements smaller than mean

my\_list = [12, 13, 16, 17, 19, 20]

new\_list = []

a = 0

for i in my\_list:

a = a + i

mean = a/len(my\_list)

for x in range(0, len(my\_list)):

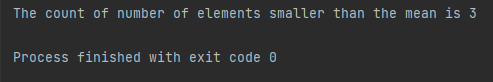
b = my\_list[x]

if b < mean:

new\_list.append(b)

print("The count of number of elements smaller than the mean is", len(new\_list))

#### Output:



# SET 2

### Correct the malformed time string, for e.g. “5:70:65” to “6:11:05”

input\_time = "5:70:65"

list\_time = input\_time.split(':')

hh = int(list\_time[0])

mm = int(list\_time[1])

ss = int(list\_time[2])

correct\_time = ""

if ss >= 60:

extra\_min = ss // 60

mm = mm + extra\_min

ss = ss % 60

if mm >= 60:

extra\_hour = mm // 60

hh = hh + extra\_hour

mm = mm % 60

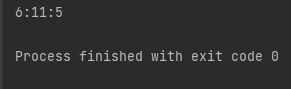
if hh >= 24:

hh = hh % 24

correct\_time = str(hh) + ":" + str(mm) + ":" + str(ss)

print(correct\_time)

#### Output:



### Correct the malformed date string, for e.g. “45/8/2018” to “14/9/2018”

months = {1: 31, 2: 28, 3: 31, 4: 30, 5: 31, 6: 30, 7: 31, 8: 31, 9: 30, 10: 31, 11: 30, 12: 31}

input\_date = input("Enter the date (dd/mm/yyyy): \n")

list\_date = input\_date.split('/')

dd = int(list\_date[0])

mm = int(list\_date[1])

yyyy = int(list\_date[2])

newDate = ""

if (yyyy % 4) == 0:

if (yyyy % 100) == 0:

if (yyyy % 400) == 0:

months[2] = 29

else:

months[2] = 28

else:

months[2] = 29

else:

months[2] = 28

c1 = dd // months[int(mm)]

rem1 = dd % months[int(mm)]

mm += c1

dd = rem1

c2 = mm // 12

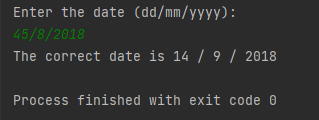
rem2 = mm % 12

yyyy += c2

mm = rem2

print('The correct date is', dd, '/', mm, '/', yyyy)

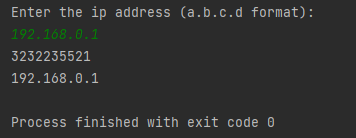
#### Output:



### Convert IP address to integer and vice versa

ip = input("Enter the ip address (a.b.c.d format): \n")  
p = 0  
int\_value = 0  
for i in ip.split('.')[::-1]:  
 int\_value += int(i) \* 256\*\*p  
 p += 1  
print(int\_value)  
  
ip\_address = ""  
for i in range(3, -1, -1):  
 num = int\_value // (256\*\*i)  
 ip\_address = ip\_address + str(num)  
 int\_value = int\_value % (256\*\*i)  
 if i > 0:  
 ip\_address = ip\_address + '.'  
print(ip\_address)

#### Output:



### Check whether the given string is isogram or not.

word\_in = input("Enter string \n")

word = word\_in.lower()

# print(word)

a = 0

char\_list = []

for char in word:

if char.isalpha():

if char in char\_list:

a += 1

char\_list.append(char)

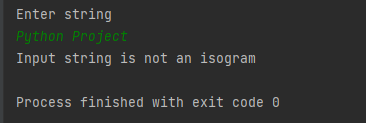
if a >= 1:

print("Input string is not an isogram")

else:

print("Input string is an isogram")

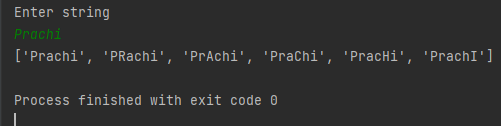
#### Output:



### Given a string, find the Mexican wave.

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

#### Output:



### Given a number, find the largest number by deleting a single digit (Order will remain same)

num = input("Enter number \n")

res = [int(x) for x in str(num)]

li = [int(x) for x in str(num)]

# print(li)

new\_list = []

while li:

mini = li[0]

for x in li:

if x < mini:

mini = x

new\_list.append(mini)

li.remove(mini)

# print(new\_list)

a = new\_list[0]

# print(a)

# print(res)

for i in range(0, len(res)):

if res[i] == a:

b = i

break

res.remove(res[b])

# print(res)

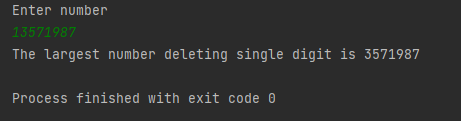
strings = [str(integer) for integer in res]

a\_string = "".join(strings)

an\_integer = int(a\_string)

print("The largest number deleting single digit is", an\_integer)

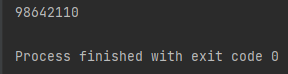
#### Output:



### Given a number, find the largest number by shuffling it.

number = 19216804  
new = []  
s = str(number)  
new[:0] = s  
new\_list = []  
while new:  
 mini = new[0]  
 for x in new:  
 if x > mini:  
 mini = x  
 new\_list.append(mini)  
 new.remove(mini)  
  
s1 = ""  
  
for x in new\_list:  
 s1 += x  
  
print(int(s1))

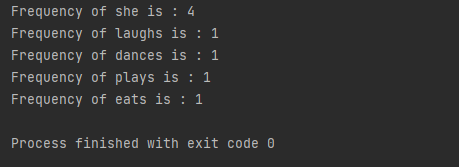
#### Output:



### Compute word frequency in given message

str = "she laughs she dances she plays she eats"  
str = str.split()  
str2 = []  
  
for i in str:  
 if i not in str2:  
 str2.append(i)  
  
for i in range(0, len(str2)):  
 print('Frequency of', str2[i], 'is :', str.count(str2[i]))

#### Output:



### Convert RGB to Hex and vice versa

#Hex to RGB

value = "FF65BA"

print("Hex input:", value)

value = value.lstrip('#')

print('RGB =', tuple(int(value[i:i+2], 16) for i in (0, 2, 4)))

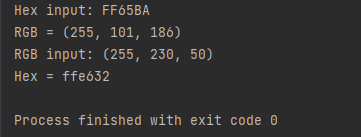
#RGB to Hex

rgb = (255, 230, 50)

print("RGB input:", rgb)

print('Hex = %02x%02x%02x' % rgb)

#### Output:



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

word = input("Enter string \n")

accum\_str = ''

r = range(len(word))

for n in r:

ch = word[n]

accum\_str += '-' + ch.upper() + ch.lower() \* n

print(accum\_str[1:])

#### Output:

