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

Code:

```
def findOddNumber(num_list):
    size=len(num_list)
    count=0
    for i in range(size):
        for j in range(size):
            if(j==i):
                continue
            if(num_list[i]==num_list[j]):
                count=1
                break
            else:
                count=0
        if(count==0):
            return num_list[i]

if __name__ == '__main__':
    size=int(input("Enter the size of the list as odd number : "))
    if((size%2)!=0):
        num_list = [int(input("Enter elements: ")) for i in range(size)]
        num = findOddNumber(num_list)
        print("Stray Number is: ", num)
    else:
        print("Wrong size entered")
```

Output Screen Shot:

```
D:\python\python.exe "C:/Users/Jagadeesh Aradhyula/Desktop/LandT/sample_python/Ass_Qn-1.py"
Enter the size of the list as odd number : 11
Enter elements: 1
Enter elements: 1
Enter elements: 2
Enter elements: 32
Enter elements: 4
Enter elements: 5
Enter elements: 4
Enter elements: 5
Enter elements: 32
Enter elements: 6
Enter elements: 2
Stray Number is:  6
```

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

```
def mean(l):
    sum=0
    m=0
    for ele in l:
        sum=sum+ele
        m=sum/(len(l))
    return m
def nearestNumber(l,m):
    templist=[]
    for i in range(len(l)):
        if ((m-l[i])==0):
            return l[i]
        templist.append(abs(m-l[i]))
    templist=sorted(templist)
    return m-templist[0]


# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    s=int(input("Enter the size of list: "))
    l=[int(input("Enter the elements: ")) for i in range(s)]
    m=mean(l)
    print("Mean for the given list: ",m)
    m=round(m)
    num=nearestNumber(l,m)
    print("Number nearest to mean: ",num)
```

Output Screen Shot:

```
D:\python\python.exe "C:/Users/Jagadeesh Aradhyula/Desktop/LandT/sample_python/main.py"
Enter the size of list: 10
Enter the elements: 15
Enter the elements: 48
Enter the elements: 987
Enter the elements: 632
Enter the elements: 45
Enter the elements: 6
Enter the elements: 58
Enter the elements: 123
Enter the elements: 3
Enter the elements: 16
Mean for the given list:  193.3
Number nearest to mean:  123
```

3. Find the Average speed of vehicle, given the speeds for fixed time intervals, e.g. [0,0.1,0.25,0.45,0.55,0.7,0.9,1.0]

```
def findAvgSpeed(speed_list):
    sum=0
    for item in speed_list:
        sum=sum+item
    return sum/len(speed_list)
```

```
if __name__ == '__main__':
    size = int(input("Enter the size of the speeds list : "))
    speed_list = [int(input("Enter speed at fixed time intervals: ")) for
i in range(size)]
    avgSpeed=findAvgSpeed(speed_list)
    print("Average Speed is: ",avgSpeed).
```

Output Screen Shot:

```
D:\python\python.exe "C:/Users/Jagadeesh Aradhyula/Desktop/LandT/sample_python/Ass_Qn-3.py"
Enter the size of the speeds list : 5
Enter speed at fixed time intervals: 10
Enter speed at fixed time intervals: 20
Enter speed at fixed time intervals: 30
Enter speed at fixed time intervals: 40
Enter speed at fixed time intervals: 50
Average Speed is:  30.0
```

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

code:

```
def numPeople(onBoard,offBoard):
    return onBoard-offBoard

if __name__ == '__main__':
    onBoard=int(input("Enter the no. of people onboarded: "))
    offBoard = int(input("Enter the no. of people alighted: "))
    num=numPeople(onBoard,offBoard)
    if(num<0):
        print("Entered Wrong Values")
    else:
        print("No. of people in Bus: ",num)
```

Output Screen Shots:

```
D:\python\python.exe "C:/Users/Jagadeesh Aradhyula/Desktop/LandT/sample_python/Ass_Qn-4.py"
Enter the no. of people onboarded: 100
Enter the no. of people alighted: 18
No. of people in Bus:  82
```

```
D:\python\python.exe "C:/Users/Jagadeesh Aradhyula/Desktop/LandT/sample_python/Ass_Qn-4.py"
Enter the no. of people onboarded: 30
Enter the no. of people alighted: 50
Entered Wrong Values
```

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

```
def findMissingNumber(original_list, modified_list):
    size=len(original_list)
    count=0
    for i in range(size):
        for j in range(size):
            if(original_list[i]==modified_list[j]):
                count=1
                break
            else:
                count=0
        if(count==0):
            return original_list[i]


if __name__ == '__main__':
    size = int(input("Enter the size of the number list : "))
    original_list = [int(input("Enter elements: ")) for i in range(size)]
    modified_list=[int(input("Enter modified elements from above list: "))
for i in range(size)]
    miss_num=findMissingNumber(original_list,modified_list)
    print("Missing Number is: ",miss_num)
```

Output Screen Shot:

```
Enter the size of the number list : 5
Enter elements: 1
Enter elements: 2
Enter elements: 3
Enter elements: 4
Enter elements: 5
Enter modified elements from above list: 1
Enter modified elements from above list: 2
Enter modified elements from above list: 4
Enter modified elements from above list: 5
Enter modified elements from above list: 6
Missing Number is:  3
```

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

Code:

```
def findLowestDiffNumber(first_list):
    first_list=sorted(first_list)
    return first_list[0]-first_list[1]


if __name__ == '__main__':
    size = int(input("Enter the size of the number list : "))
    first_list = [int(input("Enter elements: ")) for i in range(size)]
    res=findLowestDiffNumber(first_list)
    print("Difference b/w two lowest Numbers in a list  is: ",res)
```

Output Screen Shot:

```
Enter the size of the number list : 10
Enter elements: 1
Enter elements: 2
Enter elements: -100
Enter elements: -200
Enter elements: 300
Enter elements: 400
Enter elements: 500
Enter elements: 60
Enter elements: 100
Enter elements: 0
Difference b/w two lowest Numbers in a list  is:  -100
```

7. In a given list, count no.of elements smaller than their mean

Code:

```
def mean(l):
    sum=0
    m=0
    for ele in l:
        sum=sum+ele
        m=sum/(len(l))
    return m
def smallerMeanNumbersCount(l,m):
    count=0
    for i in l:
        if(i<m):
            count=count+1
    return count


# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    size=int(input("Enter the size of list: "))
    l=[int(input("Enter the elements: ")) for i in range(size)]
    mean_value=mean(l)
    print("Mean for the given list: ",mean_value)
    count=smallerMeanNumbersCount(l,mean_value)
    print("Count of Numbers smaller than mean: ",count)
```

Output Screen Shot:

```
Enter the size of list: 10
Enter the elements: 202
Enter the elements: 103
Enter the elements: 625
Enter the elements: 98
Enter the elements: 74
Enter the elements: 56
Enter the elements: 333
Enter the elements: 59
Enter the elements: 598
Enter the elements: 2
Mean for the given list:  215.0
Count of Numbers smaller than mean:  7
```

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

```
def actualTime(hour, minute, second):
    while(hour>=24 or minute>=60 or second>=60):
        if(hour>=24):
            hour=hour%24
        if(minute>=60):
            hour = hour + (minute // 60)
            minute=minute%60
        if(second>=60):
            minute = minute + (second // 60)
            second=second%60
    time=[hour,minute,second]
    return time


if __name__ == '__main__':
    hour=int(input("Enter Hour's Time: "))
    minute=int(input("Enter minute's time: "))
    second = int(input("Enter second's time: "))
    correct_time=[]
    correct_time=actualTime(hour,minute,second)
    print("Actual Time is:",end=" ")
    print(correct_time[0],":",correct_time[1],":",correct_time[2])
```

Output Screen Shot:

```
D:\python\python.exe "C:/Users/Jagadeesh Aradhyula/Desktop/LandT/sample_python/Ass_Qn-8.py"
Enter Hour's Time: 05
Enter minute's time: 70
Enter second's time: 65
Actual Time is: 6 : 11 : 5
```

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

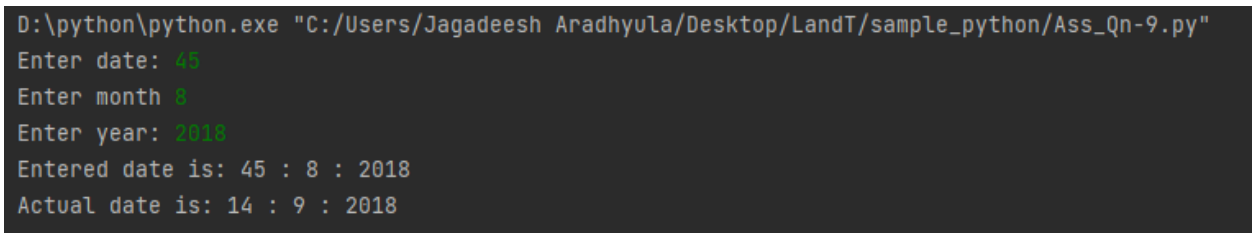
code:

```
def actualDate(date, month, year):
    while(month>12 or date>31 or (month==2 and date>29 and year%400==0) or
(month==2 and date>28 and year%400!=0)):
        year=year+(month//12)
        month=month%12
        if(month==1 or month==3 or month==5 or month==7 or month==8 or
month==10 or month==12):
            if(date>31):
                month=month+(date//31)
                date=date%31
        if (month == 4 or month == 6 or month == 9 or month == 11):
            if (date > 30):
                month = month + (date // 30)
                date = date % 30
        if(month==2 and (year%400==0)):
            if(date>29):
                month = month + (date // 29)
                date = date % 29
        else:
            if(date>28):
                month = month + (date // 28)
                date = date % 28


    date=[date,month,year]
    return date


if __name__ == '__main__':
    date=int(input("Enter date: "))
    month=int(input("Enter month "))
    year = int(input("Enter year: "))
    correct_date=[]
    correct_date=actualDate(date,month,year)
    print("Entered date is:", end=" ")
    print(date, ":", month, ":", year)
    print("Actual date is:",end=" ")
    print(correct_date[0],":",correct_date[1],":",correct_date[2])
```

Output Screen Shot:



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

Code:

```
def convertToDecimal(hexa_number):
    res=""
    for item in hexa_number:
        if item=='a':
            res=res.__add__("10.")
        if item=='b':
            res = res.__add__("11.")
        if item=='c':
            res = res.__add__("12.")
        if item=='d':
            res = res.__add__("13.")
        if item=='e':
            res = res.__add__("14.")
        if item=='f':
            res=res.__add__("15.")
    return res
def convertToHexa(decimal_number):
    res=""
    for item in decimal_number:
        if item==10:
            res=res.__add__("a.")
        if item==11:
            res=res.__add__("b.")
        if item==12:
            res = res.__add__("c.")
        if item==13:
            res = res.__add__("d.")
        if item==14:
            res=res.__add__("e.")
        if item==15:
            res=res.__add__("f.")
    return res
if __name__ == '__main__':
    hexa_number=input("Enter the ipv6 address: ")
    decimal_number=list(map(int, input("Enter a multiple value: ").split()))
    number1=convertToDecimal(hexa_number)
    print(hexa_number,"Converted to ",number1)
    number2=convertToHexa(decimal_number)
    print(decimal_number, "Converted to ", number2)
```

Output Screen Shot:

```
D:\python\python.exe "C:/Users/Jagadeesh Aradhyula/Desktop/LandT/sample_python/Ass_Qn-10.py"
Enter the ipv6 address: a.b.c.d
Enter a multiple value: 10 14 13 11
a.b.c.d Converted to  10.11.12.13.
[10, 14, 13, 11] Converted to  a.e.d.b.
```

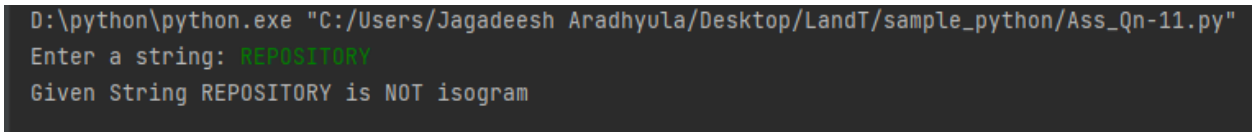
11. Check whether given string is isogram or not

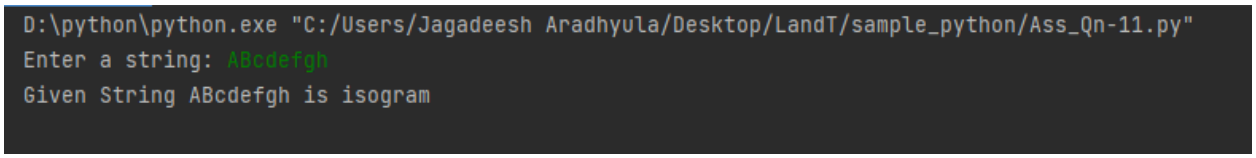
CODE:

```
def is_isogram(st):
    strlen=len(st)
    count=0
    for i in range(strlen):
        for j in range(i+1,strlen):
            if(st[i]==st[j]):
                count=count+1
                return False
    if count==0:
        return True

if __name__ == '__main__':
    st=input("Enter a string: ")
    if(is_isogram(st)):
        print("Given String",st,"is isogram")
    else:
        print("Given String",st,"is NOT isogram")
```

Output Screen Shot:





12. Given a string, find the mexican wave

Code:

```
def mexicanWave(st):
    wave=[];
    for i in range(len(st)):
        uppercase=st[i].upper()
        new_str=st[:i]+uppercase+st[i+1:]
        wave.append(new_str)
    return wave


if __name__ == '__main__':
    st=input("Enter a String: ")
    wave=mexicanWave(st)
    print(wave)
```

Output Screen Shot:

```
D:\python\python.exe "C:/Users/Jagadeesh Aradhyula/Desktop/LandT/sample_python/Ass_Qn-12.py"
Enter a String: jagadeesh
['Jagadeesh', 'jAgadeesh', 'jaGadeesh', 'jagAdeesh', 'jagaDeesh', 'jagadEesh', 'jagadeEsh', 'jagadeeSh', 'jagadeesH']
```

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

Code:

```
def removeDigit(number):
    values=[]
    for i in range(len(number)):
        new_number=number[:i]+number[i+1:]
        values.append(new_number)
    return values

def findLargerNumber(values):
    values=sorted(values)
    return values[len(values)-1]


if __name__ == '__main__':
    number=input("Enter number: ")
    values=removeDigit(number)
    print("List of number after removing one digit",values)
    largest_number=findLargerNumber(values)
    print("Largest Number among the list",largest_number)
```

Output Screen Shot:

```
D:\python\python.exe "C:/Users/Jagadeesh Aradhyula/Desktop/LandT/sample_python/Ass_Qn-13.py"
Enter number: 1234
List of number after removing one digit ['234', '134', '124', '123']
Largest Number among the list 234
```

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

Code:

```
def splitToDigits(number):
    d=[int(i) for i in number]
    return d
def findLargestNumber(digits_list):
    digits_list.sort(reverse=True)
    num=digits_list[0]
    for i in range(1,len(digits_list)):
        num=num*10+digits_list[i]
    return num


if __name__ == '__main__':
```

```
    number=input("Enter number: ")
    digits_list=splitToDigits(number)
    largest_num=findLargestNumber(digits_list)
    print("Largest number by shuffling digits in
the",number,"is:",largest_num)
```

Output Screen Shot:

```
D:\python\python.exe "C:/Users/Jagadeesh Aradhyula/Desktop/LandT/sample_python/Ass_Qn-14.py"
Enter number: 123459
Largest number by shuffling digits in the 123459 is: 954321
```

15. Compute the word frequency in given message

Code:

```
def rmvDuplicate(x):
    _size = len(x)
    repeated = []
    for i in range(_size):
        k = i + 1
        for j in range(k, _size):
            if x[i] == x[j] and x[i] not in repeated:
                repeated.append(x[i])
    return repeated


def findWordFrequency(message):
    words=message.split()
    word_list=rmvDuplicate(words)
    for item in word_list:
        print("The Frequency of ",item,"is: ",message.count(item))


if __name__ == '__main__':
    message=input("Enter message: ");
    findWordFrequency(message)
```

Output Screen Shot:

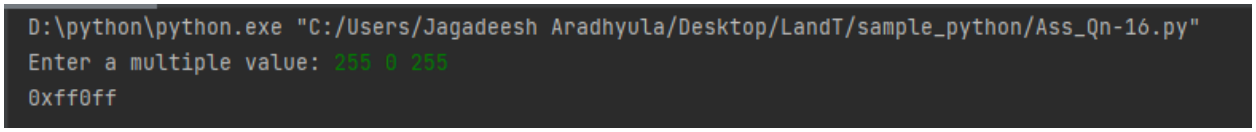
```
D:\python\python.exe "C:/Users/Jagadeesh Aradhyula/Desktop/LandT/sample_python/Ass_Qn-15.py"
Enter message: apple mango apple orange orange apple guava mango mango
The Frequency of  apple is:  3
The Frequency of  mango is:  3
The Frequency of  orange is:  2
```

16. RGB to Hex conversion and vice versa. ed. (255.0.255) into OXFF00FF

Code:

```
if __name__ == '__main__':
    RGB= list(map(int, input("Enter a multiple value: ").split()))
    str="0x"
    for i in RGB:
        str=str+hex(i)[2:]
    print(str)
```

Output Screen Shot:



17. Generate Accumulated String eg.abcd➔ A Bb Ccc Dddd

Code:

```
def accumulatedString(str):
    res=""
    for i in range(1,len(str)+1):
        res=res+str[i-1].upper()+(i-1)*str[i-1]+"-"
    return res[:len(res)-1]
if __name__ == '__main__':
    str=input("Enter string: ")
    res=accumulatedString(str)
    print("Accumulated string of ",str,"==> ",res)
```

Output Screen Shot: