**Python Assignment**

1. **To find the frequency of each word in a given message**

def freq(str):

     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]))

def main():

    str ='apple mango apple orange orange apple guava mango mango'

    freq(str)

if \_\_name\_\_=="\_\_main\_\_":

    main()

1. **RGB to Hex conversion**

def decToHexa(n):

    hexaDeciNum = ['0'] \* 100

     Counter for hexadecimal number array

    i = 0

    while (n != 0):

        temp = 0

        temp = n % 16

        if (temp < 10):

            hexaDeciNum[i] = chr(temp + 48)

            i = i + 1

        else:

            hexaDeciNum[i] = chr(temp + 55)

            i = i + 1

        n = int(n / 16)

    hexCode = ""

    if (i == 2):

        hexCode = hexCode + hexaDeciNum[0]

        hexCode = hexCode + hexaDeciNum[1]

    elif (i == 1):

        hexCode = "0"

        hexCode = hexCode + hexaDeciNum[0]

    elif (i == 0):

        hexCode = "00"

    return hexCode

def convertRGBtoHex(R, G, B):

    if ((R >= 0 and R <= 255) and

        (G >= 0 and G <= 255) and

        (B >= 0 and B <= 255)):

        hexCode = "#";

        hexCode = hexCode + decToHexa(R)

        hexCode = hexCode + decToHexa(G)

        hexCode = hexCode + decToHexa(B)

        return hexCode

    else:

        return "-1"

# Driver Code

R = 0

G = 0

B = 0

print (convertRGBtoHex(R, G, B))

R = 255

G = 255

B = 255

print (convertRGBtoHex(R, G, B))

R = 25

G = 56

B = 123

print (convertRGBtoHex(R, G, B))

R = 2

G = 3

B = 4

print (convertRGBtoHex(R, G, B))

R = 255

G = 255

B = 256

print (convertRGBtoHex(R, G, B))

1. **Given a string find Mexican wave**

def wave(str):

# Code here

result=[]

for i in range(len(str)):

str=str.lower()

temp\_list=list(str)

if temp\_list[i].isalpha():

temp\_list[i]=str[i].upper()

str=''.join(temp\_list)

result.append(str)

return result

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

def is\_isogram(word):

        clean\_word = word.lower()

    letter\_list = []

    for letter in clean\_word:

        if letter.isalpha():

            if letter in letter\_list:

                return False

            letter\_list.append(letter)

    return True

if \_\_name\_\_ == '\_\_main\_\_':

    print(is\_isogram("Machine"))

    print(is\_isogram("isogram"))

    print(is\_isogram("pretty"))

    print(is\_isogram("Alphabet "))

1. **Find out no. of people in a bus , given data of people onboarding and alighting at each station**

|  |
| --- |
| def QuestionFour(): |
|  | print("Enter the initial number of the people in the bus:") |
|  | Initial = int(input()) |
|  | print("Enter the list of number of onboarding at each station: ") |
|  | onboarding = [int(item) for item in input().split()] |
|  | print("Enter the list of number of alighting at each station: ") |
|  | alighting = [int(item) for item in input().split()] |
|  | final=[] |
|  | for i in range(0,len(onboarding)): |
|  | numberPeople = Initial+onboarding[i]-alighting[i] |
|  | final.append(numberPeople) |
|  | print("Number of people after every station is :") |
|  | print(final) |
|  | if \_\_name\_\_ == "\_\_main\_\_": QuestionFour() |

1. **Fnd the missing no. given the original list and the modified one**

|  |
| --- |
| def questionfive(): |
|  | print("Enter the original list of the numbers:") |
|  | original =[int(Item) for Item in input().split()] |
|  | print("Enter the modified list of the Items") |
|  | modified =[int(item) for item in input().split()] |
|  | for i in range(0,len(original)): |
|  | if(original[i]!=modified[i]): |
|  | print(original[i]) |
|  | break |
|  | if \_\_name\_\_ == "\_\_main\_\_":questionfive()   1. **In a given list of elements all elements are equal except one. Find out one out**  |  | | --- | | def QuestionOne(): | |  | print("Enter The list of the number") | |  | mylist = [int(item) for item in input().split()] | |  | mydictionary ={} | |  | for item in mylist: | |  | if(mydictionary.get(item)): | |  | mydictionary[item] = mydictionary.get(item)+1 | |  | else: | |  | mydictionary[item] = 1; | |  | for item in mydictionary: | |  | if(mydictionary.get(item) == 1): | |  | print("The different number is :") | |  | print(item) | |  | if \_\_name\_\_ == "\_\_main\_\_":QuestionOne() | |

1. **In a given list count no. of elements smaller than their mean**

|  |
| --- |
| def QuestionSeven(): |
|  | print("Enter the number of the list: ") |
|  | mylist = [int(element) for element in input().split()] |
|  | length = len(mylist) |
|  | total =0 |
|  | for i in mylist: |
|  | total += i |
|  | average = total/length |
|  | seclist =[] |
|  | for i in mylist: |
|  | if(i<average): |
|  | seclist.append(i) |
|  | print(seclist) |
|  | if \_\_name\_\_ == "\_\_main\_\_":QuestionSeven() |

1. **Find difference between 2 lowest no. in a list**

|  |
| --- |
| def QuestionSix(): |
|  | print("Enter the elements of the list: ") |
|  | mylist=[int(blue) for blue in input().split()] |
|  | min = mylist[0] |
|  | secondmin=9999999 |
|  | for i in mylist: |
|  | if(i==min): |
|  | continue |
|  | elif(i<min): |
|  | secondmin=min |
|  | min = i |
|  | elif(secondmin>i): |
|  | secondmin=i |
|  |  |
|  | diff = secondmin-min |
|  | print("The difference between the minimum and the second minimum is") |
|  | print(diff) |
|  | if \_\_name\_\_=="\_\_main\_\_":QuestionSix() |

1. **Find avg speed of vehicle given distance travelled for fixed time intervals**

|  |
| --- |
| def QuestionThree(): |
|  | print("Enter the list of the distance covered") |
|  | mylist = [int(item) for item in input().split()] |
|  | length = len(mylist) |
|  | Total = 0 |
|  | for i in mylist: |
|  | Total+= i |
|  | average = Total/length |
|  | print("Average speed is : ") |
|  | print(average) |
|  | if \_\_name\_\_== "\_\_main\_\_":QuestionThree() |

1. **In given list of elements , find elements which is close to its mean**

|  |
| --- |
| def QuestionTwo(): |
|  | print("Enter The list of the number") |
|  | mylist = [int(item) for item in input().split()] |
|  | total = 0 |
|  | mylisttwo=[] |
|  | for element in mylist: |
|  | total += element |
|  | lengthOfTheList = len(mylist) |
|  | average = total/lengthOfTheList |
|  | for i in range(0,len(mylist)): |
|  | mylisttwo.append(average - mylist[i]) |
|  | if(mylisttwo[i]<0): |
|  | mylisttwo[i]=mylisttwo[i]\*(-1) |
|  |  |
|  | min = mylisttwo[0] |
|  | for i in range(0,len(mylisttwo)): |
|  | if(min>mylisttwo[i]): |
|  | min=mylisttwo[i] |
|  | for i in range(0,len(mylisttwo)): |
|  | if(min==mylisttwo[i]): |
|  | print(mylist[i]) |
|  | break |
|  | if \_\_name\_\_ == "\_\_main\_\_":QuestionTwo() |

1. **Correct malformed time string**

|  |
| --- |
| def tistone(): |
|  | print("Please enter the time:") |
|  | time = input() |
|  | timelist = time.split(":") |
|  | hour = int(timelist[0]) |
|  | minutes = int(timelist[1]) |
|  | seconds = int(timelist[2]) |
|  | carry = seconds//60 |
|  | remainder = seconds%60 |
|  | minutes = minutes+carry |
|  | seconds = remainder |
|  | carry = minutes//60 |
|  | remainder = minutes%60 |
|  | hour = hour + carry |
|  | minutes = remainder |
|  | hour = hour%24 |
|  | print(hour,":",minutes,":",seconds) |
|  | if \_\_name\_\_=="\_\_main\_\_":tistone()   1. **Correct malformed date string** |
| def tisttwo(): |
|  | 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 |
|  | } |
|  | Todate = input() |
|  | currDATE = Todate.split('/') |
|  | date = int(currDATE[0]) |
|  | month = int(currDATE[1]) |
|  | year =int(currDATE[2]) |
|  | carry = date//months[int(month)] |
|  | remainder = date%months[int(month)] |
|  | month = month +carry |
|  | date = remainder |
|  | carry = month//12 |
|  | remainder = month%12 |
|  | year = year+carry |
|  | month = remainder |
|  | print(date,'/',month,'/',year) |
|  | if \_\_name\_\_=="\_\_main\_\_":tisttwo()   1. **Convert Ip address from “abcd” format to integer and vice versa**  |  | | --- | | def tistthree(): | |  | print("Enter the IP address") | |  | IP = input().split(".") | |  | IPfinal = [] | |  | for i in IP: | |  | IPfinal.append(HXtoDC(i)) | |  | print(IPfinal[0],'.',IPfinal[1],'.',IPfinal[2],'.',IPfinal[3]) | |  | def HXtoDC(hexamel): | |  | hexchart = { | |  | '1' : 1, | |  | '2' : 2, | |  | '3' : 3, | |  | '4' : 4, | |  | '5' : 5, | |  | '6' : 6, | |  | '7' : 7, | |  | '8' : 8, | |  | '9' : 9, | |  | 'a' : 10, | |  | 'b' : 11, | |  | 'c' : 12, | |  | 'd' : 13, | |  | 'e' : 14, | |  | 'f' : 15, | |  | 'A' : 10, | |  | 'B' : 11, | |  | 'C' : 12, | |  | 'D' : 13, | |  | 'E' : 14, | |  | 'F' : 15 | |  | } | |  | num=0 | |  | for i in hexamel: | |  | num \*= 16; | |  | num += hexchart[i] | |  | return num | |  | if \_\_name\_\_ =="\_\_main\_\_":tistthree() | |

1. **Given a no. , find largest no. by shuffling the digits**

|  |
| --- |
| def tlistseven(): |
|  | print("Enter the number :") |
|  | number = input() |
|  | numberlist = [] |
|  | for i in number: |
|  | numberlist.append(int(i)) |
|  | numberlist.sort() |
|  | numberlist.reverse() |
|  | large='' |
|  | for i in numberlist: |
|  | large=large+str(i) |
|  | print(large) |
|  | if \_\_name\_\_ =="\_\_main\_\_":tlistseven() |

1. **Given a no. , find the largest no. by deleting a single digit**

|  |
| --- |
| def tlistsix(): |
|  | print("Enter a number :") |
|  | number = input() |
|  | length = len(number) |
|  | for i in range(0,len(number)-1): |
|  |  |
|  | if(int(number[i])<int(number[i+1])): |
|  | number = number[:i]+number[i+1:] |
|  | break |
|  | if(length == len(number)): |
|  | number=number[:-1] |
|  | print(number) |
|  | if \_\_name\_\_ == "\_\_main\_\_":tlistsix() |