UNIT – 4

LESSON 1 AND 2:

*# str1="This is my first String"  
# print(str1)  
# print(str1[11])  
# print(str1[-6])  
  
# a="HELLO"  
# print(a[0:4])  
# print(a[:3])  
# print(a[::])  
#  
# # print(a[::-1]) reversing the string  
# print(a[4:1:-1])  
  
# a="Python"  
# print(a[4:1:-1])  
  
  
# str="How are you?"  
# print("String is",str)  
# print(str[4:7])  
# print(str[2:5:1])  
# print(str[-4:-1:])  
# print(str[-2:-5:-1])  
# print(str[-4:])  
  
  
# a=str(input(""))  
# if len(a)<3:  
# print(a)  
# else:  
# x=a[0:2]  
# y=a[-2:]  
# z=x+y  
# print(z)  
  
# a=str(input(""))  
# print(a[1:-1])  
  
# a=str(input(""))  
# if len(a)==1:  
# print(a)  
# elif len(a)==0:  
# print("Null")  
# else:  
# x=a[0]  
# y=a[-1]  
# z=a[1:-1]  
# print(y+z+x)  
  
  
# a=str(input(""))  
# b=str(input(""))  
# print(a+b+b+a) Concatenation  
# print(a+b+b[::-1]+a[::-1]) Reverse Concat  
  
  
# a=str(input("Enter the string: "))  
# b=int(input("Enter the index you want to remove: "))  
# x=a[0:b]  
# y=a[b+1:] remove the entered index  
# print(x+y)*

*# a=str(input(""))  
# b=str(input(""))  
# if a>1 and b>1:  
# x=a[1:]  
# y=b[1:]  
# print(x+y)  
# else:  
# print("null")*

a=str(input(**""**))  
print(a\*4)  
print(4\*a[::-1])

a=str(input(**""**))  
**if** len(a)>3:  
 print(a[0:3]\*3)  
**else**:  
 print(a)

a=str(input(**"String: "**))  
b=int(input(**"Num: "**))  
print(a\*b)

a=str(input(**"String: "**))  
b=int(input(**"num: "**))  
**if** b<=len(a) **and** b>0:  
 print(a[:b]\*b)  
**else**:  
 print(**"num should be positive, less than length of string"**)

str1=**"Hello"  
del** str1 deleting a string using **'del'**print(str1)

IN BUILT STRING METHODS:

1. capitalize()- Capitalize the first letter   
of a string  
  
2. upper()- Convert the entire string into  
uppercase  
  
3. lower()- Convert the entire string into   
lowercase  
  
4. title()- Convert the first alphabet of every  
word into uppercase  
  
5. swapcase()- Upper to lower, Lower to upper

6. split()- Spliting the string into a list items  
ex- str=**"This is my title test"**print(a.split())  
=> [**'This'**,**'is'**,**'my'**,**'title'**,**'test'**]

a=**"this is my title test"**print(a.split(**"m"**))  
  
a=**"10,20,30,40"**print(a.split(**","**))

7. center(width,**"fillchar"**)- Function used to pad a string  
a=**"hello"**print(a.center(11,**"&"**))  
=>&&&hello&&&

8. count()- Counts the frequency of   
a particular substring **in** a string

a=**"Welcome to Python Class, Welcome"**print(a.count(**"Welcome"**))

9. replace(old,new)- Replaces all the old substrings  
**with** new substrings **and if** old substring **is not** present  
no changes are made

a=**"java is simple java"**b=**"Python"**print(a.replace(**"java"**,b))

10. join()- Joins strings **with** the elements of iterable

b=**'.'**a=[**'www'**,**'codetantra'**,**'com'**]  
print(b.join(a))

11. isupper()- Use to check whether a string **is** uppercase  
**or not**

a=**"Python is simple"**print(a.isupper())

12. isupper()- Use to check whether a string **is** lowercase  
**or not**a=**"Python is simple"**print(a.islower())

13. isalpha()- Used to check whether the string **is** alphabet  
**or not**a=**"hello programmer"**print(a.isalpha())  
b=**"hello23good"**print(b.isalpha())  
c=**"ennjebejc"**print(c.isalpha())

14. isalnum()- used to check **for** alpha numeric characters  
a=**"alpha789"**print(a.isalnum())  
b=**'alpha'**print(a.isalnum())  
c=**'1234'**print(a.isalnum())  
d=**"wcj32!1"**print(a.isalnum())

15. isdigit()- used to check whether all the character are  
digits **or not**

a=**"123"**b=**"123hello"**print(a.isdigit())  
print(b.isdigit())

16. isspace()- used to check whether every characters are   
spaces **or not**

17. istitle()- used to check the first chara b cter of every   
word **is** capital **or not**

Uses of backslash(\)  
print(**'It\'s very powerful'**)  
print(**"Hello \"everyone"**)

**if** we want to print \t **and** \n **as** it **is**we us repr()

str=**"Hello\tPython\nPython"**print(str)  
print(repr(str))

a=str(input(**""**))  
b=str(input(**""**))  
**if** len(a)>len(b):  
 print(a+b+a)  
**elif** len(b)>len(a):  
 print(b+a+b)  
**elif** len(a)==len(b):  
 print(**"Strings are same length"**)

18. startswith(substring)- used to check whether a   
string **is** starting **with** the given substring  
a=**"hello world"**print(a.startswith(**"h"**))

19. endswith(substring)- used to check whether a string  
**is** ending **with** the given substring  
a=**"hello world"**print(a.endswith(**"d"**))

*returns -1 id substring is not found*

20. find(substring)- used to find the index of the first   
occurence of the substring

a=**"hello python hello"**print(a.find(**"he"**))

21. min(a)- used to find the minimum unicode of the   
string  
  
  
22. max(a)- used to find the maximum unicode of the string

a=str(input(**"str: "**))  
**if** (a.startswith(**"Python"**)) **and** (a.endswith(**"Programming"**)):  
 print(**"valid"**)  
 print(**"Character with min value:"**,min(a))  
 print(**"Character with max value:"**,max(a))  
**else**:  
 print(**"Not valid"**)

String Modules:

**import** string  
print(string.punctuation)  
print(string.digits)  
print(string.printable)

a=str(input(**""**))  
b=str(input(**""**)) count the substrings **in** the string  
print(a.count(b))

a=str(input(**""**))  
i=0  
**while** i<len(a):  
 print(a[i]\*2,end=**""**)  
 i+=1

a=str(input(**""**))  
**if** len(a)%2==0:  
 s=len(a)/2  
 s=int(s)  
 print(a[0:s])  
**else**:  
 s = len(a) / 2  
 s = int(s)  
 print(a[s+1:])

24/11/22:

a=str(input(**""**))  
**if** len(a)==0:  
 print(**"str:"**)  
 print(**"null"**)  
 print(**"first:"**)  
 print(**"second"**)  
 print(**"Original:"**)  
**elif** len(a)==1:  
 print(**"str:"**,a)  
 print(a) alternate print  
 print(**"first:"**,a)  
 print(**"second:"**)  
 print(**"Original:"**)  
**else**:  
 fstr=**""** sstr=**""  
 for** i **in** range(0,len(a)):  
 **if** i%2==0:  
 fstr=fstr+a[i]  
 **else**:  
 sstr=sstr+a[i]  
 print(**"first:"**,fstr)  
 print(**"second:"**,sstr)  
 print(**"Original:"**,a)

a=str(input(**""**))  
x=**"" incremental print  
for** i **in** range(0,len(a)+1):  
 x=x+a[0:i]  
print(x)

23. isprintable()- It returns true if all the characters are printable or else false

a=**"Python \n"**print(a.isprintable())

24. isalpha()-It returns true **if** all the characters are alphabets **or not**

**import** string  
print(**"Character\t ASCII Code"**)  
**for** x **in** string.ascii\_letters:  
 print(x,**"\t\t\t"**,ord(x))

a=str(input(**""**))  
b=sorted(a)  
l1=[]  
**for** char **in** b:  
 print(**"'{0}'\t{1}"**.format(char,b.count(char)))  
 l1.append(char)  
print(l1)

#display the frequency of each character and print in a list

a=str(input(**""**))  
b=sorted(a)  
l1=[]  
**for** i **in** b:  
 **if** i **not in** l1:  
 print(**"'{0}'\t{1}"**.format(i,b.count(i)))  
 l1.append(i)  
print(l1)

#display the frequency of each number and print in a list

Lists:

sequence=(1,2,3,4,5)  
l1=list(sequence)  
print(l1)  
print(type(l1))

creating a list using list()

a=str(input(**""**))  
print(type(a))  
b=a.split()  
print(b)  
print(type(b))

#taking string as input and splitting and adding each item in a list

Very important

l1=[1,2,3]  
l2=[3,2,1]  
print(l1==l2)

* False

Using the comparison operator

a=str(input(**""**))  
b=a.split(**","**)  
print(**"data:"**,a) creating a list **from** string **and** printing the entered index  
print(**"list:"**,b)  
x=int(input(**"Enter index:"**))  
**if** x<len(b) **and** x>=-(len(b)):  
 print(b[x])  
**else**:  
 print(**"invalid"**)

list1=[**"hi"**,**"hello"**,**"Lists"**]  
**for** i **in** range(0,len(list1)):  
 print(list1[i],end=**" "**)  
print()  
list1[2]=**"Python"  
for** i **in** range(0,len(list1)):  
 print(list1[i],end=**" "**)  
print()  
list1.append(**"Code is life"**)  
**for** i **in** range(0,len(list1)):  
 print(list1[i],end=**" "**)  
 print()  
list1.extend([45,67,89])  
**for** i **in** range(0,len(list1)):  
 print(list1[i],end=**" "**)

a=str(input(**""**))  
b=a.split(**","**)  
**if** b[0]==**"3" or** b[-1]==**"3"**:  
 print(**"True"**)  
**else**:  
 print(**"False"**)

a=[1,2,3,4,5]  
a[0:3]=[100,100,100]  
print(a)

UPDATING MULTIPLE VALUES USING SLICING APPROACH

a=[1,2,3,4,5]  
a[0:0]=[20,30,45]  
print(a) inserting on first index using empty slice

a=str(input(**""**))  
b=a.split(**","**)  
x=int(input(**"Enter Index:"**))  
**if** x<len(b) **and** x>-(len(b)):  
 y=str(input(**"Enter the updation:"**))  
 b[x]=y  
 print(b)  
**else**:  
 print(**"Invalid"**)

a=str(input(**"Enter the numbers with comma(,): "**))  
b=a.split(**","**)  
c=max(b)  
print(b)  
print(**"Largest:"**,c,**"Last:"**,b[-1])

Slide number 43 :-

29/11/2022

a=[1,2,3,4,5,6]  
b=a  
print(b **is** a)  
  
a[0]=100  
print(b)

CLONING  
a=[1,2,3,4,5,6]  
b=a[:]  
print(b)  
print(b **is** a)

a[0]=100  
print(a)  
print(b)

dlist=[1,2,3,4,5,6,7]  
**del** dlist[3]  
print(dlist)  
**del** dlist[2:6]  
print(dlist)  
**del** dlist  
print(dlist)

pop()- used to remove a particular element from a list and it also returns the removed element

remove()- used to remove a particular element from a list and if we delete the entire list, it would be still available in the memory

del()- used to delete a particular element from a list and if we delete the entire list, it would show us error as it is no longer available in the memory

RESUME FROM SLIDE NUMBER 56

a=str(input(**""**))  
b=a.split(**","**)  
print(b)  
i=0  
c=[]  
**while** i<len(b):  
 **if** b[i] **not in** c:  
 c.append(b[i])  
 **else**:  
 **pass** i+=1  
print(c)

LIST FUNCTIONS:

all()-

print(all([**" "**,**","**,**"1"**,**"2"**]))

any()

print(any([**" "**,**","**,**"1"**,**"2"**]))  
print(any([[]]))  
print(any([0,0,0,0]))

enumerate()-

print(list(enumerate([**"a"**,**"b"**,**"c"**,**"d"**])))

len()-

list()-

print(list(**"abcdef"**))  
print(list((**"a"**,**"b"**,**"c"**,**"d"**,**"e"**,**"f"**)))

max()-

print(max([1,2,3,4,5]))

min()-

print(min([1,2,3,4,5]))

a=str(input(**""**)).split(**","**)  
**for** i **in** range(len(a)):  
 a[i]=int(a[i])  
print(**"min:"**,min(a))  
print(**"max:"**,max(a))  
print(**"difference:"**,max(a)-min(a))

a=str(input(**""**)).split(**","**)  
b=str(input(**""**)).split(**","**)  
i=0  
str1=**"{"  
if** len(a)!=len(b):  
 print(**"Lists are of different lengths"**)  
**else**:  
 **while** i<len(a):  
 str1=str1+**"'"**+a[i]+**"'"**+**":"**+**"'"**+b[i]+**"'"**+**","** i+=1  
 print(str1[0:len(str1)-1]+**"}"**)

*# a=str(input(""))  
# x=a.split(",")  
# b=str(input(""))  
# y=b.split(",")  
# l1=[]  
# for i in range(len(x)):  
# x[i]=int(x[i])  
# for j in range(len(y)):  
# y[i]=int(y[i])  
# if len(x)==len(y):  
# for i in range(0,len(x)):  
# l1.append(abs(x[i]-y[i]))  
# print(l1)*x=[**"a"**,**"b"**,**"c"**,**"d"**]  
x.insert(0,1)  
print(x)  
x.insert(len(x),**"e"**)  
print(x)  
x.insert(1,**"e"**)  
print(x)

a=str(input(**"data: "**))  
c=int(input(**"Element: "**))  
b=a.split(**","**)  
i=0  
x=0  
**for** i **in** range(len(b)):  
 b[i]=int(b[i])  
**if** c **not in** b:  
 print(**"{} occurs 0 times"**.format(c))  
**else**:  
 **for** i **in** range(0,len(b)):  
 **if** b[i]==c:  
 x=x+1  
 i+=1  
 print(**"{} occurs {} times"**.format(c,x))

a=str(input(**"data: "**))  
c=int(input(**"Element: "**))  
b=a.split(**","**)  
i=0  
x=0  
**for** i **in** range(len(b)):  
 b[i]=int(b[i])  
**if** c **not in** b:  
 print(**"{} occurs 0 times"**.format(c))  
**else**:  
 **for** i **in** range(0,len(b)):  
 **if** b[i]==c:  
 **if** b[i+1]==c:  
 print(**"True"**)  
 **else**:  
 print(**"False"**)  
 **break** i+=1

1/12/2022:

a=str(input(**"Upper: "**))  
b=str(input(**"Lower: "**))  
i=ord(**"A"**)  
j=ord(**"a"**)  
alpha=[]  
**for** k **in** range(26):  
 alpha.append(chr(i))  
 alpha.append(chr(j))  
 i+=1  
 j+=1  
print(alpha)

a=str(input(**""**))  
b=a.split(**","**)  
**for** i **in** range(len(b)):  
 b[i]=int(b[i])  
x=sorted(b)  
**if** b==x:  
 print(**"True"**)  
**else**:  
 print(**"False"**)

a=str(input(**"Please enter a sentence: "**))  
b=a.lower()  
x=**""  
for** i **in** range(0,len(b)):  
 **if** b[i]!=**" "**:  
 x=x+b[i]  
x=sorted(x)  
printed=[]  
**for** char **in** x:  
 **if** char **not in** printed:  
 **if** char.isalpha():  
 print(char,**"\t"**,x.count(char))  
 printed.append(char)

LISTS OVER……

TUPLES:

a=[1,2,3,4,5,6,7]  
a=tuple(a)  
print(a)

t1=(1,2,3,[4,5],5)  
t1[3][0]=42 list updation supported  
print(t1)

t1[0]=100 tuple updation not supported  
print(t1)

tuple1=(**"hi"**,**"hello"**,**"7"**,**"5"**)  
print(tuple1)  
print(type(tuple1))

converting into list:

list1=list(tuple1)  
print(list1)  
print(type(list1))

mytuple=(**"i"**,**"love"**,**"Python"**)  
print(**"Given Tuple:"**,mytuple)  
list1=list(mytuple)  
print(**"After converting Tuple into List:"**,list1)  
list1[1]=**"practice"**print(**"After updating the index 1:"**,list1)  
print(**"After converting back to Tuple:"**,tuple(list1))

a=(20,40,60,**"apple"**,**"ball"**)  
t1=(1,)  
print(type(t1))  
t2=(1)  
print(type(t2))

a=str(input(**"data: "**))  
b=a.split(**","**)  
c=tuple(b)  
print(**"List:"**,b)  
print(**"Tuple:"**,c)  
x=int(input(**"index: "**))  
**if** x<len(b) **and** x>=-(len(b)):  
 print(c[x])  
**else**:  
 print(**"invalid"**)

mytuple=(**"this"**,10.0,**"is"**,**"float"**,3.6)  
print(mytuple[0])  
print(mytuple[1])  
print(mytuple[-1]) SLICING  
print(mytuple[0:])  
print(mytuple[0:-1])  
print(mytuple[::-1])

a=str(input(**"data1: "**))  
b=a.split(**","**)  
b=tuple(b)  
x=int(input(**"value: "**))  
print(**"tuple \* {} = {}"**.format(x,(b\*x)))  
y=str(input(**"data2: "**))  
c=y.split(**","**)  
c=tuple(c)  
print(**"concatenation:"**,b+c)

a=str(input(**"data: "**))  
b=a.split(**","**)  
b=tuple(b)  
print(**"tuple: "**,b)  
c=str(input(**"to check? "**))  
**if** c **in** b:  
 print(**"True"**)  
**else**:  
 print(**"False"**)

mytup=(1,2,3,4,5,[6,7,8])  
**del** mytup[5][2]  
print(mytup)

*#6/12/2022*

a=str(input(**"data: "**)).split(**","**)  
i=int(input(**"Index: "**))  
print(**"tuple: "**,tuple(a))  
**if** i!=-1:  
 **if** i<len(a) **and** i>=-(len(a)):  
 x=a[0:i]  
 y=a[i+1:]  
 print(x+y)  
 **else**:  
 print(**"Index out of range"**)  
**else**:  
 print(**"after removal: "**,tuple(a[:-1]))

a=str(input(**"data: "**)).split(**","**)  
b=str(input(**"Element: "**))  
**if** b **in** a:  
 print(**"Before deletion:"**, tuple(a))  
 x=a.remove(b)  
  
 print(**"After deletion: "**,tuple(a))  
**else**:  
 print(**"Enter existing element"**)

a=str(input(**"Data: "**)).split(**","**)  
x=int(input(**"Start index: "**))  
y=int(input(**"End index: "**))  
**if** (x<len(a)**and** y<len(a))**and**(x>=-(len(a))**and** x>=-(len(a))):  
 print(**"Tuple in range: "**,tuple(a[x:y]))  
**else**:  
 print(**"Enter valid index"**)

a=str(input(**"Data: "**)).split(**","**)  
b=str(input(**"Element: "**))  
**if** b **in** a:  
 x=a.count(b)  
 print(**"Existed {} time(s)"**.format(x))  
**else**:  
 print(**"Enter valid element"**)

a=str(input(**"Data: "**)).split(**","**)  
tup=tuple(a)  
print(**"Tuple:"**,tup)  
element=input(**"Element: "**)  
**if** element **in** a:  
 index=tup.index(element)  
 print(**"Index:"**,index)  
**else**:  
 print(**"Enter an element that exists in the tuple"**)

DICTIONARY:

month={**"Jan"**:1,**"Feb"**:2,**"Mar"**:3}  
print(month)  
print(type(month))  
  
  
a=dict([(**'one'**,1),(**"two"**,2)])  
print(a)  
print(type(a))  
  
  
print(month[**"Jan"**])

a={**"USA"**:**"Washington"**,**"India"**:**"New Delhi"**,**"Nepal"**:**"Kathmandu"**}  
print(a[**"India"**]) key error  
print(a[**"Sri lanka"**])

a={**"USA"**:**"Washington"**,**"India"**:**"New Delhi"**,**"Nepal"**:**"Kathmandu"**}  
print(a.get(**"Aus"**,**"Not available"**)) overwriting **"None"**

dict1={**"name"**:**"Jay"**,**"number"**:514,**"age"**:14}  
**for** i **in** dict1:  
 print(i,dict[i])  
  
 *# Getting the total elements of dictionary***for** i,j **in** dict1.items():  
 print(i,j)  
print(dict1.keys())  
print(dict1.values())

Updating the elements of a value using key

month={**"Jan"**:1,**"Feb"**:2,**"Mar"**:3}  
print(month)  
month[**"Jan"**]=100  
print(month)

a=str(input(**"Data: "**)).split(**","**)  
b=str(input(**"Data2: "**)).split(**","**)  
c=dict(zip(a,b))  
print(**"List1:"**,a)  
print(**"List2:"**,b)  
print(**"Dictionary:"**,sorted(c.items()))  
print(list(zip(a,b)))  
print(set(zip(a,b)))  
print(tuple(zip(a,b)))  
print(dict(zip(a,b)))  
r=zip(a,b)  
print(r)

a=str(input(**"Data1: "**)).split(**","**)  
b=str(input(**"Data2: "**)).split(**","**)  
x=dict()  
**if** len(a)==len(b):  
 **for** i **in** range(len(a)):  
 x[a[i]]=b[i]  
 print(x)  
 print(sorted(x.items()))  
  
**else**:  
 print(**"Length should be equal"**)

a=str(input(**"Data1: "**)).split(**","**)  
b=str(input(**"Data2: "**)).split(**","**)  
c=dict(zip(a,b))  
x=input(**"Key: "**)  
**if** x **in** c:  
 print(**"Value: "**,c[x])  
**else**:  
 print(**"Value: None"**)

Checking for element in dictionary and printing it

a=str(input(**"Data1: "**)).split(**","**)  
b=str(input(**"Data2: "**)).split(**","**)  
c=dict(zip(a,b))  
x=input(**"Key: "**)  
**if** x **in** c:  
 print(**"True"**)  
**else**:  
 print(**"False"**)

only checking for the presence of element in a dictionary

a=str(input(**"Data1: "**)).split(**","**)  
b=str(input(**"Data2: "**)).split(**","**)  
c=dict(zip(a,b))  
**for** key,value **in** sorted(c.items()):  
 print(key,value)

a=str(input(**"Data1: "**)).split(**","**)  
b=str(input(**"Data2: "**)).split(**","**)  
c=dict(zip(a,b))  
**for** key,value **in** sorted(c.items()):  
 print(key,**"->"**,value)

fruits={1:**"apple"**,2:**"orange"**,3:**"mango"**,4:**"grapes"**}  
print(type(fruits))  
  
print(fruits.pop(4))  
print(fruits)  
  
print(fruits.popitem())  
print(fruits)  
  
**del** fruits[2]  
print(fruits)  
  
fruits.clear()  
print(fruits)  
  
**del** fruits  
print(fruits)

a=str(input(**"Data1: "**)).split(**","**)  
b=str(input(**"Data2: "**)).split(**","**)  
c=dict(zip(a,b))  
k=input(**"Key: "**)  
v=input(**"Value: "**)  
**if** k **in** c:  
 x=c.pop(k)  
 print(c)  
**else**:  
 print(**"Key does not exist"**)

a=str(input(**"Data1: "**)).split(**","**)  
b=str(input(**"Data2: "**)).split(**","**)  
a=sorted(a)  
b=sorted(b)  
c=dict(zip(a,b))  
print(**"Min:"**,b[0])  
print(**"Max:"**,b[-1])

a=str(input(**"Data1: "**)).split(**","**)  
b=str(input(**"Data2: "**)).split(**","**)  
c=dict(zip(a,b))  
k=input(**"Key: "**)  
v=input(**"Value: "**)  
c[k]=v  
print(c)

8/12/2022:

dict1={1:**"alpha"**,2:**""**,**""**:**"Gamma"**}  
print(all(dict1))

all() function in dictionary

dict1={1:**"alpha"**,2:**""**,**""**:**"Gamma"**}  
print(any(dict1))

all() function in dictionary

a=str(input(**"Data1: "**)).split(**","**)  
b=str(input(**"Data2: "**)).split(**","**)  
x=dict(zip(a,b))  
d3=dict()  
**for** key **in** x:  
 d3[x[key]]=key  
print(sorted(x.items()))  
print(sorted(d3.items()))

SWAPPING THE KEYS AND VALUES IN A DICTIONARY

dict1={(**"Cleese"**,**"John"**):[1,2,3],  
 (**"Chapman"**,**"Graham"**):[4,5,6],  
 (**"Idle"**,**"Eric"**):[7,8,9],  
 (**"Jones"**,**"Terry"**):[10,11,12],  
 (**"Gilliam"**,**"Terry"**):[13,14,15,16,17,18],  
 (**"Palin"**,**"Michael"**):[19,20]}  
**for** last,first **in** sorted(dict1):  
 print(first,last,dict1[last,first])

SWAPPING THE VALUES INSIDE THE KEYS

dict1={**"a"**:100,**"b"**:200,**"c"**:300}  
dict2={**"a"**:300,**"b"**:200,**"d"**:400}  
dict3={}  
**for** key **in**(list(dict1.keys())+list(dict2.keys())):  
 dict3[key]=dict1.get(key,0)+dict2.get(key,0)  
print(sorted(dict3.items()))

ADDING THE VALUES OF COMMON KEYS

a=str(input(**"Data1: "**)).split(**","**)  
b=str(input(**"Data2: "**)).split(**","**)  
c=str(input(**"Data3: "**)).split(**","**)  
d=str(input(**"Data4: "**)).split(**","**)  
e=str(input(**"Key: "**))  
x=dict(zip(a,b))  
y=dict(zip(c,d))  
**if** e **in** x.keys() **and** e **not in** y.keys():  
 print(**"Present in First Dictionary"**)  
**elif** e **in** y.keys() **and** e **not in** x.keys():  
 print(**"Present in Second Dictionary"**)  
**elif** e **not in** y.keys() **and** e **not in** x.keys():  
 print(**"Not present in any dictionary"**)  
**else**:  
 print(**"Present in both dictionaries"**)

Checking the presence of a particular key in dictionaries

LIST COMPREHENSION:

l2=[x\*x **for** x **in** range(1,11) **if** x%2==0]  
print(l2)

n=int(input(**""**))  
n=10  
print([**"even" if**(n%2==0) **else "odd"**])

a=str(input(**""**))  
x=[]  
print([i **for** i **in** a])

*#13/12/2022*

*#print([a\*b for a in [1,2,3] for b in [10,20,30]])  
#nested list comprehension*

print([a **for** a **in** [10,8,5,4] **for** b **in** [4,7,5,10] **if** a==b])  
Printing the commmon elements **in** two lists

print([a **for** a **in** range(50) **if** a%2==0 **if** a%3==0])  
printing the elements **from** 0 to 50 which are divisible by 2 **and** 3

matrix=[[1,2,3,4],[5,6,7,8],[9,10,11,12]]  
transposed=[]  
**for** i **in** range(len(matrix[0])):  
 transposed.append([row[i] **for** row **in** matrix])  
print(transposed)  
transpose of matrix using single list comprehension  
  
  
  
  
transposed\_double=[]  
transposed\_double=[[row[i] **for** row **in** matrix] **for** i **in** range(len(matrix[0]))]  
print(transposed\_double)  
transpose of matrix using double list comprehension

14/12/2022:

Dictionary Comprehension:

l1=[10,20,30]  
dict1={key:value **for** key,value **in** enumerate(l1)}  
print(dict1)

dict2={i:i\*\*2 **for** i **in** range(1,11)}  
print(dict2)

dict1={k:**"python" for** k **in "Codetantra"**}  
print(dict1)

dict2={i.lower(): i **for** i **in "PYTHON"**}  
print(dict2)

dict1={x:chr(x) **for** x **in** range(65,91)}  
print(dict1)  
  
  
  
dict2={x:y **for** y,x **in** dict1.items()}  
print(dict2)

**def** sparseMatrix(sparseMatrix, m, n):  
   
 size = 0  
 **for** i **in** range(m):  
 **for** j **in** range(n):  
 **if** (sparseMatrix[i][j] != 0):  
 size += 1  
  
 rows, cols = (3, size)  
 compressMatrix = [[0 **for** i **in** range(cols)] **for** j **in** range(rows)]  
  
 k = 0  
 **for** i **in** range(m):  
 **for** j **in** range(n):  
 **if** (sparseMatrix[i][j] != 0):  
 compressMatrix[0][k] = i  
 compressMatrix[1][k] = j  
 compressMatrix[2][k] = sparseMatrix[i][j]  
 k += 1  
  
 print(**"Sparse representation:"**)  
 **for** i **in** compressMatrix:  
 print(i)  
  
  
  
m=int(input(**"Enter row size: "**))  
n=int(input(**"Enter column size: "**))  
print(**"Elements:"**)  
mat=[[int(input()) **for** x **in** range (n)] **for** y **in** range(m)]  
print(**"Sparse matrix is: "**)  
**for** i **in** range(m):  
 **for** j **in** range(n):  
 print(mat[i][j],end=**" "**)  
 print()  
sparseMatrix(mat,m,n)

UNIT 4 ENDS…..