**PYTHON ASSIGNMENT**

**SUBMITTED BY : SHIVANI SHARMA**

**TO : ANIKET DOKANIA**

**Q1) Create a table with a large number of records (you can find it with a google search or use this**

**link - https://github.com/datacharmer/test\_db). Use MySQL Database.One can setup MySQL on**

**localhost. Write some basic queries using python. Suppose you want to process/fetch a large**

**number of records using python while keeping your memory usage low. Think of approaches on**

**how to accomplish this and Implement.**

**Hint: Use Generator**

from \_\_future\_\_ import generators

import mysql.connector

def ResultIter(cursor, arraysize = 1000 ):

while True:

results = cursor.fetchmany(arraysize)

if not results:

break

for result in results:

yield result

mydb = mysql.connector.connect (

host = "localhost",

user = "root" ,

port = "3306" ,

database = "student"

)

cursor = mydb.cursor()

cursor.execute('select \* from mydb')

for result in ResultIter(cursor):

print(result)

============================================================= RESTART: C:\Users\Admin\Documents\dbcode.py =============================================================

('ruhi', 23)

('sam', 45)

**(Q2)** **Define a class Person and its two child classes: Male and Female. All classes have a**

**method &quot;get\_gender&quot; which can print &quot;Male&quot; for Male class quot;Female&quot; for Female.**

class Person(object):

def getGender( self ):

return "Unknown"

class Male( Person ):

def getGender( self ):

return "Male"

class Female( Person ):

def getGender( self ):

return "Female"

aMale = Male()

aFemale= Female()

print (aMale.getGender())

print (aFemale.getGender())

========================= RESTART: C:\Python38\Q2 ).py =========================

Male

Female

>>>

**Q3) With a given list [12,24,35,24,88,120,155,88,120,155], write a program to print this list**

**after removing all duplicate values with original order reserved.**

**Hint: Use set() to store a number of values without duplicates.**

l=[12,24,35,24,88,120,155,88,120,155]

s=set(l)

res=[]

for i in l:

if(i in s):

res.append(i)

s.remove(i)

print(res)

========================== RESTART: C:/Python38/dup.py =========================

[12, 24, 35, 88, 120, 155]

>>>

(**Q4) Write a program to generate a 3\*5\*8 3D array whose each element is 0.**

array = [[ [0 for col in range(8)] for col in range(5)] for row in range(3)]

print (array)

===================== RESTART: C:\Python38\binarysearch.py =====================

[[[0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0]], [[0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0]], [[0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0]]]

>>>

**(Q5) Write a binary search function which searches an item in a sorted list. The function**

**should return the index of element to be searched in the list.**

from bisect import bisect\_left

def BinarySearch(a, x):

i = bisect\_left(a, x)

if i != len(a) and a[i] == x:

return i

else:

return -1

a = [1, 2, 4, 4, 8]

x = int(4)

res = BinarySearch(a, x)

if res == -1:

print(x, "is absent")

else:

print("First occurrence of", x, "is present at", res)

========================= RESTART: C:\Python38\shiv.py =========================

First occurrence of 4 is present at 2

>>>

**(Q6)Write a program using generator to print the numbers which can be divisible by 5**

**and 7 between 0 and n in comma separated form while n is input by console.**

nl=[]

for x in range(700, 3000):

if (x%7==0) and (x%5==0):

nl.append(str(x))

print (','.join(nl))

=================== RESTART: C:/Python38/input by console.py ===================

700,735,770,805,840,875,910,945,980,1015,1050,1085,1120,1155,1190,1225,1260,1295,1330,1365,1400,1435,1470,1505,1540,1575,1610,1645,1680,1715,1750,1785,1820,1855,1890,1925,1960,1995,2030,2065,2100,2135,2170,2205,2240,2275,2310,2345,2380,2415,2450,2485,2520,2555,2590,2625,2660,2695,2730,2765,2800,2835,2870,2905,2940,2975

>>>

**(Q7) Assuming that we have some email addresses in the &quot;username@companyname.com&quot;**

**format, write a program to print the company name of a given email address. Both user**

**names and company names are composed of letters only.**

import re

email = "shivani@datagrokr.com sam@company.com "

pattern = "\w+@(\w+).com"

ans = re.findall(pattern , email)

print(ans)

=================== RESTART: C:/Python38/input by console.py ===================

['datagrokr', 'company']

**(Q8) Write a program which can map() to make a list whose elements are square of numbers**

**between 1 and 20.**

li = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]

squaredNumbers = map(lambda x: x\*\*2, li)

print(list(squaredNumbers))

=================== RESTART: C:/Python38/input by console.py ===================

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400]

>>>

**(Q9) Implement a program dir\_tree.py that takes a directory as argument and prints all the files**

**in that directory recursively as a tree.**

import os

rootDir = '.'

for dirName, subdirList, fileList in os.walk(rootDir):

print('Found directory: %s' % dirName)

for fname in fileList:

print('\t%s' % fname)

**Q10) Write an iterator class ReverseIter, that takes a list and iterates it from the reverse**

**Direction.**

class Reverselter:

def \_\_init\_\_(self,l):

for i in reversed(range(len(l))):

print(l[i])

l=[1,2,3,4,5]

Reverselter(l)

**Q11)Write a program anti\_html.py that takes a URL as argument, downloads the html from web**

**and print it after stripping html tags.**

import urllib.request , urllib.error , urllib.parse

url = 'enter url here'

response = urllib.request.urlopen(url)

webcontent = response.read()

print(re.sub('<[^<]+?>','',str(webcontent)))