COMPUTER SCIENCE PRACTICAL FILE

Submitted by: Vinayak Bector
Class: XII D
Roll No: 33
Session:
School:

INDEX

S. No.	Name of Program	Page No.	Date	Sign
	Take a sample of ten phishing e-mails			
	WAP to count the words “to” and “the” present in a text file “Poem.txt”			
	WAP to remove all the lines that contain the article `a’			
	Write a function DISPLAYWORDS( )			
	WAP to read characters from the keyboard one by one till user enters “end”			
	WAP that reads a text file and creates another file that is identical except that every sequence of consecutive blank spaces is replaced by a single space			
	WAP that reads a text file file1.txt and appends at the end of another file file2.txt, every line from file1.txt preceded by a line number.			
	WAP to find the size of file in bytes, number of words and number of lines			
	Create a binary file with name and roll number. Search for a given roll number and display the name, if not found display appropriate message.			
	Create a binary file with roll number, name and marks. Input a roll number and update the marks.			
	Write a menu driven telephone directory program			
	Create a binary file and write a string having two lines in it			
	write a method in python to search and display all the content in a pickled file			
	WAP to increase the salary by Rs 2000/- of the employee having empno as 1251 in the file emp1.dat			

TERM 1

P1. Take names and marks of 5 students and save as key: value pairs in a dictionary RESULT. WAP that prints the dictionary contents in ascending order of marks. (Do not use built in methods)

In [4]:

```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
#1
def bubbleSort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1] :
                arr[j], arr[j+1] = arr[j+1], arr[j]
d1 = {}
l1 =[]
for i in range(1,4):
    key = input("Enter NAME of student  ")
    l1.append(key)
    value = input("Enter MARKS of student  ")
    d1[key]= value
bubbleSort(l1)
for k in l1:
    print("Key",k)
    print('Value',d1[k])
```

```
Date:  2020-06-29 21:09:04.962657
UserName:  jovyan
Enter NAME of student  q
Enter MARKS of student  1
Enter NAME of student  w
Enter MARKS of student  2
Enter NAME of student  e
Enter MARKS of student  3
Key e
Value 3
Key q
Value 1
Key w
Value 2
```

P2. WAP to print the following pattern:

In [71]:

```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
#2
z=""
for row in range(0,7):
    for column in range(0,7):
        if row == 0 or row == 6:
            if column >= 0 and column <= 6:
                z+="*"
        elif row+column==6:
            z+="*"
        else:
            z+=" "
    z+="\n"
print(z)
```

```
Date:  2020-06-27 20:35:08.376627
UserName:  jovyan
*******
      *
     *
    *
   *
  *
 *
*******
```

P3. WAP to generate a 3 X 4 X 6 3D array whose each element is *.

In [74]:

```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
#3

l1 = []
l2 = []
l3 = []

for k in range(6):
    l1.append('*')
for j in range(4):
    l2.append(l1)
for i in range(3):
    l3.append(l2)

print(l3)
```

```
Date:  2020-06-27 20:40:51.209059
UserName:  jovyan
[[['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*',
'*', '*', '*', '*']], [['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*',
'*', '*'], ['*', '*', '*', '*', '*', '*']], [['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*'],
['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*', '*']]]
```

Functions

P4. Write a function to calculate volume of a box with appropriate default values for its parameters. Your function should have the following input parameters:

a) Length of box

b) Width of box

c) Height of box Test it by writing complete program to invoke it.

In [9]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))

#4
def volume(l = 1.0,b = 1.0,h =1.0):
    return(l*b*h)
l = float(input('Length of the Box'))
b = float(input('Width of the Box'))
h = float(input('Height of the Box'))
volume(l,b,h)
```

```
Date:  2020-06-27 19:41:19.871500
UserName:  jovyan
Length of the Box2
Width of the Box3
Height of the Box5
```

Out[9]: 30.0

P5. Write a program to have the following functions: a) A function that takes a number as argument and calculates cube for it. The function does not return a value. If there is no value passed to the function in function call, the function should calculate cube of 2

b) A function that takes two char arguments and returns True if both the arguments are equal otherwise false.

Test both these functions by giving appropriate function call statements.

In [12]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
#5a
def cube(a = ""):
    if a =="":
        return(2**3)
    else:
        return(a**3)

a = int(input('Please enter the first value '))
cube(a)
```

```
Date:  2020-06-27 19:42:20.655434
UserName:  jovyan
Please enter the first value 7
```

Out[12]: 343

In [13]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
#5b
def equality(h,k):
    if h == k:
        return True
    else:
        return False
h = input('Please enter the first value ')
k = input('Please enter the first value ')
equality(h,k)
```

```
Date:  2020-06-27 19:42:32.272443
UserName:  jovyan
Please enter the first value hero
Please enter the first value hero
```

Out[13]: True

P6. Write a Python function to check whether a number is perfect or not (number to be tested as parameter). Test the function by giving appropriate function call statements.

In [17]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
#6
def factors(x):
    l1 = []
    sm = 0
    for i in range(1,round(x/2)+1):
        if x%i == 0:
            l1.append(i)
    for i in l1:
        sm +=i
    if sm == x:
        print('Perfect Number')
    else :
        print('Not a Perfect Number')
x = int(input('Enter a number to check whether it is Perfect or not '))
factors(x)
```

```
Date:  2020-06-27 19:44:30.863069
UserName:  jovyan
Enter a number to check whether it is Perfect or not 6
Perfect Number
```

P7. Write a Python program to execute a string containing Python code.

In [55]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
#7
def execute(s):
    try:
        x = exec(s)
    except Exception as e:
        print("Error:", e)

stg = str(input('Enter the code which you want to execute'))
execute(stg)
```

```
Date:  2020-06-27 20:12:01.707308
UserName:  jovyan
Enter the code which you want to executeprint('Hello World')
Hello World
```

P8. Write a function that takes a number n and then returns a randomly generated number having exactly n digits(not starting with zero) e.g., if n is 2 then function can randomly return a number 10 – 99 but 07, 02 are not valid two digit numbers.

In [69]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
import random

#8
def digit(n):
    try :
        range_start = 10**(n-1)
        range_end = (10**n)-1
        return random.randint(range_start, range_end)
    except Exception as SyntaxError:
        print("Numbers can't start with 0 ")

    print("The number of digits in the number are:",count)
n = int(input('Enter the number of digits '))
digit(n)
```

```
Date:  2020-06-27 20:25:44.526651
UserName:  jovyan
Enter the number of digits 5
```

Out[69]: 60891

P9. Write a program that generates a series using a function which takes first and last values of the series and then generates four terms that are equidistant .g., if two numbers passed are 1 and 7 then function returns 1 3 5 7.

In [64]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
#9
def series(a,b):
    diff = (a+b)/4
    return (a, a+diff, b-diff, b)

t = int(input('enter your first number of the series'))
k = int(input('enter your last number of the series'))
series(t,k)
```

```
Date:  2020-06-27 20:22:34.968847
UserName:  jovyan
enter your first number of the series1
enter your last number of the series7
```

Out[64]: (1, 3.0, 5.0, 7)

P10. Write a function that takes one argument (a positive integer) and reports if the argument is prime or not. Write a program that invokes this function.

In [31]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
#10
def prime(p):
    for i in range(2,round(p/2)):
        if p %i == 0:
            print("Not prime")

        else:
            print('Prime')
            break

p = int(input("Enter the number to test primality "))
prime(p)
```

```
Date:  2020-06-27 19:52:01.774155
UserName:  jovyan
Enter the number to test primality 7
Prime
```

Recursion

P11. Do program 10 using recursive technique.

In [36]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
#11
def Isprime(N,a=2):
    if N == 1:
        print('Neither Prime nor composite')
    elif N == 2:
        return False
    elif N%2 == 0:
        return False
    elif a>=N :
        return True

    else:
        return Isprime(N,a+1)

N = int(input('Enter number to test primality '))
Isprime(N)
```

```
Date:  2020-06-27 19:54:26.261618
UserName:  jovyan
Enter number to test primality 7
```

Out[36]:
```
True
```

P12. Implement a function product( ) to multiply 2 positive numbers recursively using addition.

In [72]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))

def prdt(a,b):
    if b == 1:
        return a
    elif b ==0:
        return 0
    else:

        return a + prdt(a,b-1)

a = int(input('1st number '))
b = int(input('2nd number '))
prdt(a,b)
```

```
Date:  2020-06-27 20:37:34.639046
UserName:  jovyan
1st number 3
2nd number 4
```

Out[72]: 12

P13. Write recursive code to compute GCD of two numbers.

In [2]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
def gcd(a,b):

    if a < b:
        lo = a
        hi = b
    else:
        lo = b
        hi = a

    if hi%lo == 0:
        print(lo)
    elif lo == 1:
        print("1")
    else:
        gcd(hi,hi%lo)


a = int(input('1st Number  '))
b = int(input('2nd Number  '))
gcd(a,b)
```

```
Date:  2020-06-29 21:29:39.470157
UserName:  jovyan
1st Number  8
2nd Number  12
4
```

P14. The hailstone sequence starting at a positive integer n is generated by following 2 simple rules. If n is even, the next number in the sequence is n/2. If n is odd, the next number in the sequence is 3*n + 1. Repeating this process, the hailstone sequence gets generated. Write a recursive function hailstone(n) which prints the hailstone sequence beginning at n. Stop when the sequence reaches number 1.

In [48]:

```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
#13
l2=[]
def hailstone(n):
    if n == 1:
        l2.append(1)
        print(l2)
    elif n == 0:
        return 0
    elif n%2 == 0:
        l2.append(n)
        n = (n/2)
        hailstone(n)
    elif n%2 != 0:
        l2.append(n)
        n = 3*n +1
        hailstone(n)

n = int(input('Plase enter the number to start the sequence'))
hailstone(n)
```

```
Date:  2020-06-27 20:05:09.821940
UserName:  jovyan
Plase enter the number to start the sequence5
[5, 16, 8.0, 4.0, 2.0, 1]
```

P15. WAP that takes a number and checks if it is a happy number by using the following two functions in it:

sum_sq_digits(x) : returns the sum of the square of the digits of number x using recursion
ishappy( ) : checks if the number is happy number by calling the function sum_sq_digits and displays an appropriate message.

In [15]:

```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))

def sum_sq_digits(n):
    caln = 0;
    while(n):
        caln += (n % 10) * (n % 10);
        n = int(n / 10);
    return caln;
def ishappy(n):
    if n//10 == 0 and n == 1:
        print("HappyNumber")
    elif n//10 == 0 and n !=1:
        print('Not a happy number')
    else:
        key = sum_sq_digits(n)
        ishappy(key)
n = int(input("Enter Value to be checked "))
ishappy(n)
```

```
Date:  2020-06-29 22:00:57.787055
UserName:  jovyan
Enter Value to be checked 28
HappyNumber
```

P16. A list namely Adm stores admission numbers of 100 students in it, sorted in ascending order of admission numbers (integers). WAP that takes an admission number and looks for it in list Adm using recursive binary search technique.

```
In [13]:
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))

def binary(arr,num,start,end):
    m = (start+end)//2
    if arr[m] > num:
        binary(arr,num,start,m)
    elif arr[m]< num:
        binary(arr,num,m,end)
    elif arr[m]== num:
        print("Number was found at position ",m+1)
    elif start>=end:
        print('NOT found')
add= [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32,
k = int(input("Enter the addmission number "))
binary(add,k,0,len(add))
```

```
Date:  2020-06-29 21:50:57.924576
UserName:  jovyan
Enter the addmission number 14
Number was found at position  14
```

Using Python Libraries

P17. Write a program with following function: remove_letter(sentence, letter) This function returns a copy of the above string with every instance of the indicated letter removed. E.g. , remove_letter("Welcome Everyone", "e") should return the string "Wlcom vryon".

```
In [1]:
def remove_letter(s,l):
    k = s.replace(l, "")
    print(k)

string= str(input("Please enter your string  "))
letter =str(input("Please enter your letter  "))
remove_letter(string,letter)
```

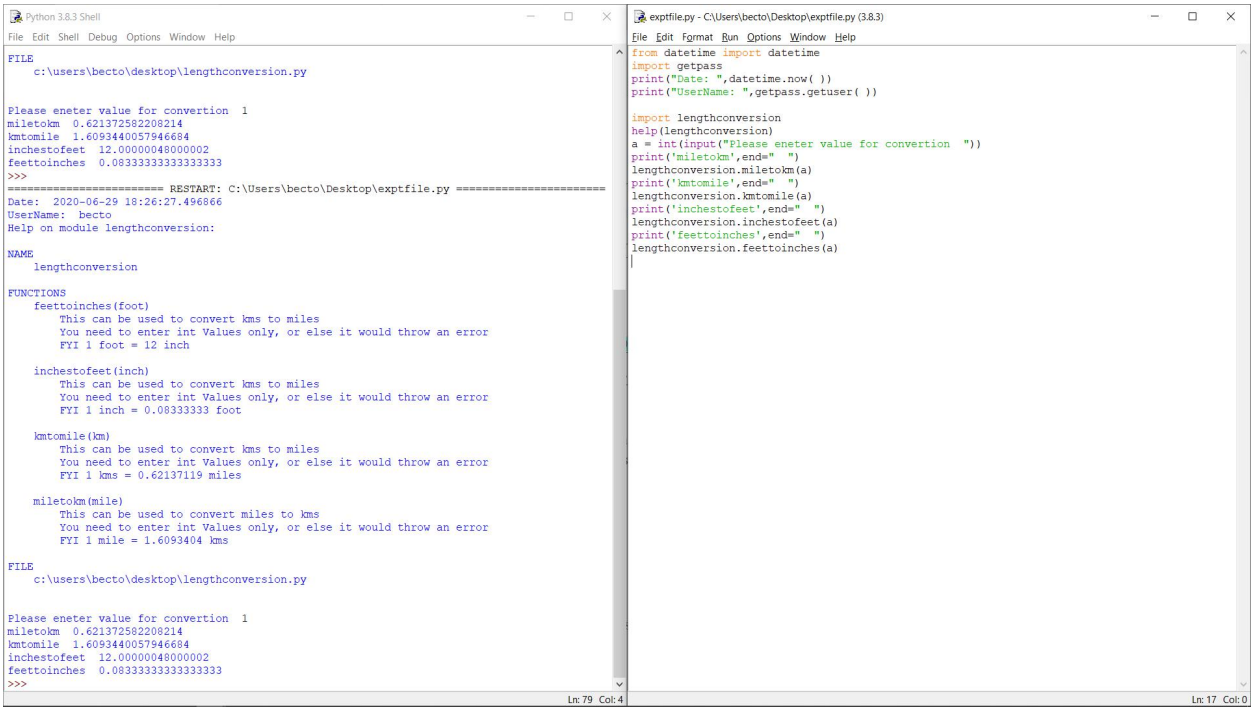
```
Please enter your string  Hello World
Please enter your letter  l
Heo Word
```

```
In [ ]:
```

P18. Create a module lengthconversion.py that stores functions for various length conversions:

miletokm( ) kmtomile( ) feettoinches( ) inchestofeet( )

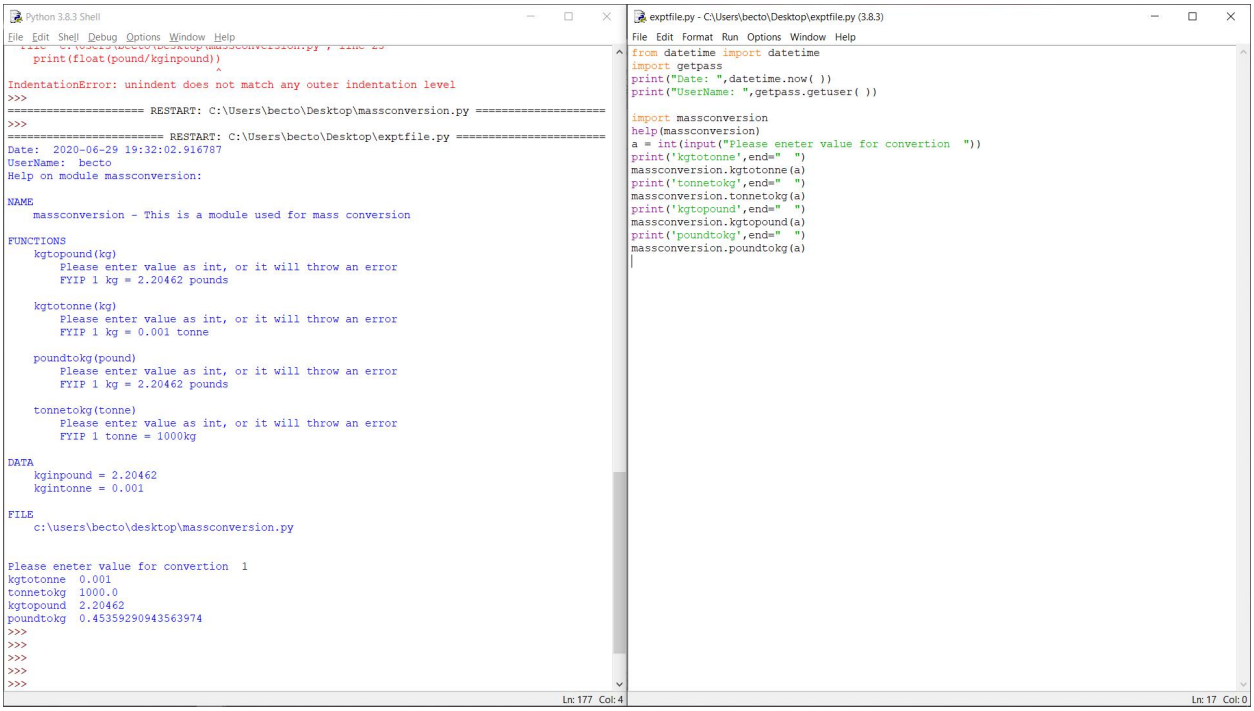
Also store the constants – mileinkm = 1.609344, feetininches = 12. help( ) method should display proper information (to be shown as output)



P19. Create a module massconversion.py that stores functions for various mass conversions:

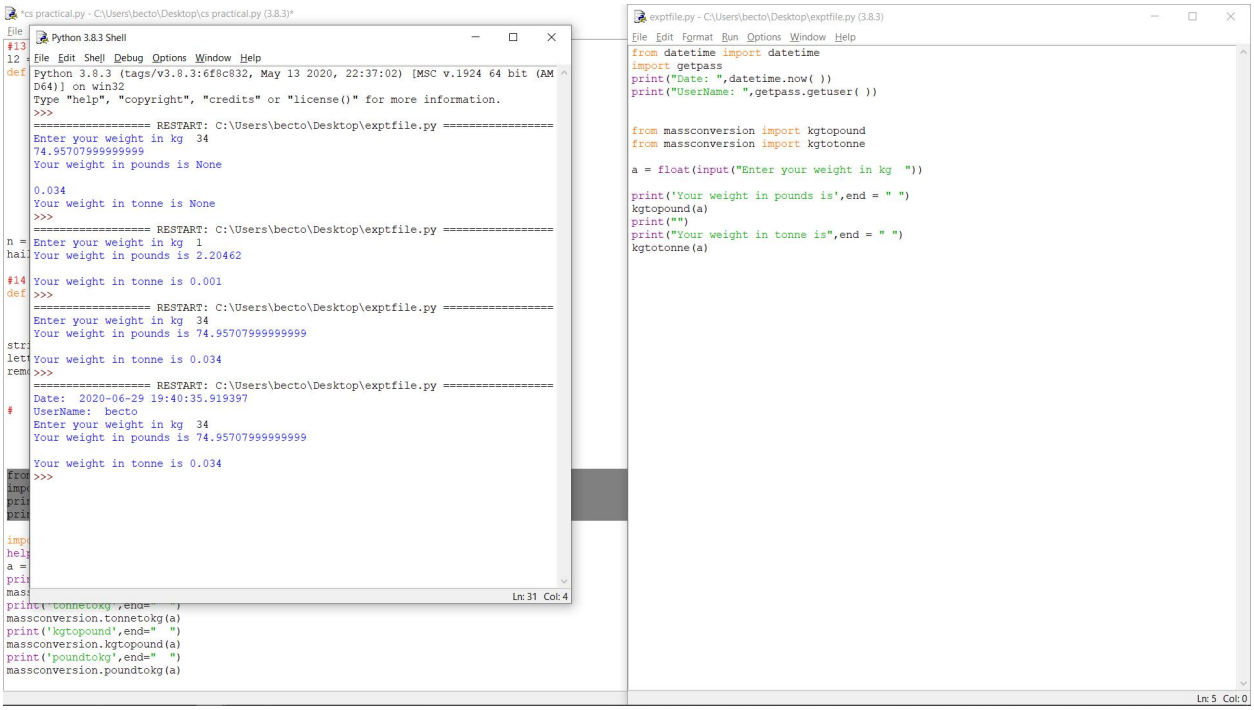
kgtotonne( ) tonnetokg( ) kgtopound( ) poundtokg( )

Also store the constants – kgintonne = 0.001, kginpound = 2.20462. help( ) method should display proper information (to be shown as output)



P20. Create a package from the above two modules as this:

Make sure that above packages meet the requirements of being a Python Package. Access at least two methods of lengthconversion.py using import statement. Access at least two methods of massconversion.py using from-import statement.



Interface Python with MySQL

P21. Write a menu-driven program to do the following tasks: 1. Add a record 2. Add multiple records 3. Display all records 4. Update a record 5. Delete a record 6. Exit

Instructions for the above program:

a) The program should ask the user for the choice number (1 to 6) every performed operation. The program should display the menu continually till the user enters 'no' as choice.

b) In every operation, the respective function should be called. So there should be 5 functions in total, each function should include creating a connection with database, performing the respective operation and closing the connection. c) For all the operations, input should be taken from the user. d) In update, insert and delete operations, you must use commit( ) and rollback( ).

e) Create the following table EMPLOYEE in your MySQL database: (Columns are: ENAME, DEPT_NAME, DESIGNATION, SALARY, DATE_OF_JOINING)

```
import mysql.connector as sqlcon
import sys

l1 = ['1. Add a record','2. Add multiple records',' 3. Display all records',' 4. Update a record',' 5. Delete a record',' 6. Exit']
for k in range(6):
    print(l1[k])
z =1
while(z):
    i = int(input('Enter number from 1 to 6   '))
    if i == 1:
        con1 = sqlcon.connect(host = 'localhost', user = 'root',passwd ='80A1D715F7EE',database='school' )
        if con1.is_connected():
            print("CONNECTED")
        print('ADD A RECORD')
        cursor1 = con1.cursor()
        ENAME = input("ENTER NAME")
        DEPT_NAME = input("ENTER DEPT_NAME")
        DESIGNATION = input("ENTER DESIGNATION")
        SALARY = int(input("ENTER SALARY"))
        DATE_OF_JOINING = input("ENTER DATE_OF_JOINING")
        query1 = "INSERT INTO EMPLOYEE VALUES('{}','{}','{}',{},{})".format(ENAME, DEPT_NAME, DESIGNATION, SALARY, DATE_OF_JOINING)
        cursor1.execute(query1)
        con1.commit()
        con1.close()
    elif i == 2:
        print('ADD MULTIPLE RECORDS')
        con2 = sqlcon.connect(host = 'localhost', user = 'root',passwd ='80A1D715F7EE',database='school' )
        if con2.is_connected():
            print("CONNECTED")
        cursor2 = con2.cursor()
        n = int(input('How many records do you want to submit'))
        l1=[]
        for i in range(n):
            t=()
            ENAME = input("ENTER NAME")
            DEPT_NAME = input("ENTER DEPT_NAME")
            DESIGNATION = input("ENTER DESIGNATION")
            SALARY = int(input("ENTER SALARY"))
            DATE_OF_JOINING = input("ENTER DATE_OF_JOINING")
            t =(ENAME, DEPT_NAME, DESIGNATION, SALARY, DATE_OF_JOINING)
            l1.append(t)
        query2 = "INSERT INTO EMPLOYEE VALUES(%s,%s,%s,%s)"
        values = l1
        cursor2.executemany(query2,values)
        con2.commit()
        con2.close()
    elif i == 3:
        print("DISPLAY ALL RECORDS")
```

```
        con1.close()
    elif i == 2:
        print('ADD MULTIPLE RECORDS')
        con2 = sqlcon.connect(host = 'localhost', user = 'root',passwd ='80A1D715F7EE',database='school' )
        if con2.is_connected():
            print("CONNECTED")
        cursor2 = con2.cursor()
        n = int(input('How many records do you want to submit'))
        l1=[]
        for i in range(n):
            t=()
            ENAME = input("ENTER NAME")
            DEPT_NAME = input("ENTER DEPT_NAME")
            DESIGNATION = input("ENTER DESIGNATION")
            SALARY = int(input("ENTER SALARY"))
            DATE_OF_JOINING = input("ENTER DATE_OF_JOINING")
            t =(ENAME, DEPT_NAME, DESIGNATION, SALARY, DATE_OF_JOINING)
            l1.append(t)
        query2 = "INSERT INTO EMPLOYEE VALUES(%s,%s,%s,%s)"
        values = l1
        cursor2.executemany(query2,values)
        con2.commit()
        con2.close()
    elif i == 3:
        print("DISPLAY ALL RECORDS")
        con3 = sqlcon.connect(host = 'localhost', user = 'root',passwd ='80A1D715F7EE',database='school' )
        if con3.is_connected():
            print("CONNECTED")
        cursor3 = con3.cursor()
        query3 = 'SELECT * FROM EMPLOYEE'
        cursor3.execute(query3)
        result = cursor3.fetchall()
        for row in result:
            print(row)

        con3.close()
    elif i == 4:
        print('UPDATE A RECORD')
        con4 = sqlcon.connect(host = 'localhost', user = 'root',passwd ='80A1D715F7EE',database='school' )
        if con1.is_connected():
            print("CONNECTED")
        cursor4 = con4.cursor()
        column_name = input("Enter column_name")
        new_value = input('Enter new_value')
        clause = str(input('Enter where clause'))
        query4 = "UPDATE EMPLOYEE SET '{}' = {} WHERE ={}".format(column_name,new_value,clause)
        cursor4 = execute(query4)
        con4.commit()
        con4.close()
    elif i == 5:
```

```
        cursor3.execute(query3)
        result = cursor3.fetchall()
        for row in result:
            print(row)

        con3.close()
    elif i == 4:
        print('UPDATE A RECORD')
        con4 = sqlcon.connect(host = 'localhost', user = 'root',passwd ='80A1D715F7EE',database='school' )
        if con1.is_connected():
            print("CONNECTED")
        cursor4 = con4.cursor()
        column_name = input("Enter column_name")
        new_value = input('Enter new_value')
        clause = str(input('Enter where clause'))
        query4 = "UPDATE EMPLOYEE SET '{}' = {} WHERE ={}".format(column_name,new_value,clause)
        cursor4 = execute(query4)
        con4.commit()
        con4.close()
    elif i == 5:
        print('DELETE A RECORD')
        con5 = sqlcon.connect(host = 'localhost', user = 'root',passwd ='80A1D715F7EE',database='school' )
        if con5.is_connected():
            print("CONNECTED")
        cursor5 = con5.cursor()
        clause = str(input("Enter Where clause "))
        query5 = "DELETE FROM EMPLOYEE WHERE ={}".format(clause)

        cursor5.execute(query5)
        con5.commit()
        con5.close()
    elif i == 6:
        z= False
        break
    else:
        continue
```

```
+--------+------------+-------------+--------+------------------+
| ENAME  | DEPT_NAME  | DESIGNATION | SALARY | DATE_OF_JOINING  |
+--------+------------+-------------+--------+------------------+
| ANAND  | SALES      | SALESMAN    |  11000 | 0000-00-00       |
| ANIL   | SALES      | SALESMAN    |  16000 | 0000-00-00       |
| GEORGE | SALES      | CLERK       |   9500 | 0000-00-00       |
| MILIND | ACCOUNTING | CLERK       |  13000 | 0000-00-00       |
| SAXENA | SALES      | SALESMAN    |  12500 | 0000-00-00       |
| TOMAR  | SALES      | SALESMAN    |  15000 | 0000-00-00       |
| TOMMAR | SALES      | SALESMAN    |  14500 | 0000-00-00       |
+--------+------------+-------------+--------+------------------+
7 rows in set (0.01 sec)
```

```
Date:  2020-06-30 05:06:13.782827
UserName:  becto
1. Add a record
2. Add multiple records
 3. Display all records
 4. Update a record
 5. Delete a record
 6. Exit
Enter number from 1 to 6   3
DISPLAY ALL RECORDS
CONNECTED
('ANAND', 'SALES', 'SALESMAN', 11000, None)
('ANIL', 'SALES', 'SALESMAN', 16000, None)
('GEORGE', 'SALES', 'CLERK', 9500, None)
('MILIND', 'ACCOUNTING', 'CLERK', 13000, None)
('SAXENA', 'SALES', 'SALESMAN', 12500, None)
('TOMAR', 'SALES', 'SALESMAN', 15000, None)
('TOMMAR', 'SALES', 'SALESMAN', 14500, None)
Enter number from 1 to 6   6
>>>
```

Write a program in Python to do the following:

a) Take Doctor_Id and Hospital_Id from the user and display all details from the respective tables

b) Display the list of doctors with their details for a given speciality (taken as input from user)

c) Display list of doctors with their details within a given hospital

```
import mysql.connector as sqlcon
import sys
con1 = sqlcon.connect(host = 'localhost', user = 'root',passwd ='80A1D715F7EE',database='school' )
if con1.is_connected():
    print("CONNECTED")
Doctor_id = int(input("Doctor Id "))
query1 = "SELECT DOCTOR_NAME FROM DOCTORS WHERE DOCTOE_ID = {}".format(Doctor_id)
con1.execute(query1)
special = input("Enter the speciality of the doctor")
query2 = "SELECT DOCTOR_NAME FROM DOCTORS WHERE SPECIALITY = {}".format(special)
cursor1.execute(query2)
result = cursor1.fetchall()
for row in result:
    print(row)
    con1.execute(query1)
HOSPITAL_NAME = input("Enter the name of the hospital")
query2 = "SELECT DOCTOR_NAME FROM DOCTORS WHERE HOSPITAL_ID = (SELECT HOSPITAL_ID FROM HOSPITAL WHERE HOSPITAL_NAME = '{}')".format(HOSPITAL_NAME)
cursor1.execute(query2)
result = cursor1.fetchall()
for row in result:
    print(row)
```

Q17 WAP to read a text file line by line and display each word separated by a #.

Q17

In [1]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q17():
    with open('file.txt','r') as fin :
        print("#".join((fin.read()).split()))
q17()
```

```
Date:  2020-09-11 04:03:18.428220
UserName:  jovyan

Hello#World!#Python#this#side,#hope#you're#having#a#good#day.
```

Q18 Read a text file and display the number of vowels/ consonants/uppercase/ lowercase characters in the file.

Q18

In [7]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q18():
    with open('file.txt','r') as fin :
        count_vowels = count_cons = count_upper = coun_lower= 0
        for i in fin :
            for j in i :
                if j in ['a','e','i','o','u','A','E','I','O','U']:
                    count_vowels +=1
                elif j.isupper():
                    count_upper +=1
                elif j.islower():
                    coun_lower +=1
                elif j in ['0','1','2','3','4','5','6','7','8','9']:
                    continue
                else:
                    count_cons +=1
    print("count vowels",count_vowels)
    print("count cons",count_cons)
    print("count upper",count_upper)
    print("coun lower",coun_lower)
q18()
```

```
Date:  2020-09-11 04:07:59.964959
UserName:  jovyan

count vowels 18
count cons 14
count upper 3
coun lower 26
```

Q19 WAP to count the words “to” and “the” present in a text file “Poem.txt”

Q19

In [14]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q19():
    with open('Poem.txt','r') as fin:
        count_to = count_the =0
        for i in (fin.read()).split():
            if i == "the":
                count_the +=1
            elif i =="to":
                count_to +=1
            else:
                continue
        print("count to",count_to, " \t count the",count_the)
q19()
```

```
Date:  2020-09-11 04:21:52.911315
UserName:  jovyan

count to 24      count the 39
```

Q22 WAP to take the details of books from the user (title, price for each book) as many as he/she wants and write the records in text file. (e.g. of one record: Computer Science with Python, 500)

Q22

In [19]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q22():
    limit = int(input("Enter the number of records to be entered "))
    with open('file.txt','a') as fout:

        for i in range(limit):
            title = input("Enter title of the book")
            price = int(input("Price of the book"))
            fout.write(title+ str(price))
q22()
```

```
Date:  2020-09-11 04:29:00.515808
UserName:  jovyan

Enter the number of records to be entered 2
Enter title of the book1Book
Price of the book1
Enter title of the book2Book
Price of the book2
```

Q23. Write a function DISPLAYWORDS( ) in python to read lines from the text file POEM.TXT and display only those words, which are less than 4 characters.

Q23

In [22]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q23_displaywords():
    with open("Poem.txt","r") as fin:
        for i in (fin.read()).split():
            if len(i)<4:
                print(i, end="  ")
q23_displaywords()
```

```
Date:  2020-09-11 04:33:43.321221
UserName:  jovyan

his  in  his  His  who  or  him  but  who  him  who  was  his  his  was  his  For  he  had  at  to  to  all  be  of  and  He
had  at  and  his  he  had  not  as  a  but  as  a  The  man  was  He  Dr  do  so.  But  if  a  man  in  be  On  the  if  hi
m  as  a  he  The  in  a  It  was  for  to  to  any  He  it  be  if  the  man  be  by  the  but  by  who  did  not  He  to
two  of  his  any  to  his  no  the  of  the  had  to  the  of  the  and  Hr  had  no  for  the  and  he  his  the  a  man
to  his  He  the  was  to  him  the  He  not  a  to  he  a  he  him  and  not  His  to  get  him  the  of  the  old  did  no
t  the  he  in  a  to  in  he  was  a  The  of  it  had  to  in  He  the  It  was  a  to  be  at  war  he  the  of  the  It
was  and  He  it  he  his  He  is  why  he  not  the  man  All  the  and  The  old  the  not  to  the  of  He  the  man  to
he  was  the  sea  him  and  him  her  If  the  the  gun  and  the  sea  had  on  as  as  his  the  as  as  be  to  put  him
the  But  of  was  to  put  him  the  if  the  man  was  he  him  to  the  but  he  was  the  not  him  to  the  He  not  th
e  man  he  had  the  of  Toy  is  a  toy  for  6th  to  8th  The  in  the  two  of  The  be  by  the  Dr.  HOD  at  IIT  is
of  the  are  and  to  all  the  on  to  to  by  5th
```

24.WAP to read characters from the keyboard one by one till user enters “end” to finish the program. All lower case characters get stored in file LOWER.TXT, upper case in UPPER.TXT and all other characters in the file OTHERS.TXT.

Q24

In [25]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q24():
    with open("Lower.txt","w") as fout1:
        with open("Upper.txt","w") as fout2:
            with open("Other.txt","w") as fout3:
                while True:
                    x = input("Enter your string  ")
                    if x == 'end':
                        break
                    else :
                        for i in x:
                            if i.isupper():
                                fout2.write(i+',')
                            elif i.islower():
                                fout1.write(i+',')
                            else:
                                fout3.write(i+',')
q24()
```

```
Date:  2020-09-11 04:40:53.743676
UserName:  jovyan

Enter your string  Hi bro
Enter your string  How are you?
Enter your string  Hope doing great!
Enter your string  OKay, bye.
Enter your string  end
```

jupyter Lower.txt a minute ago

File Edit View Language

```
1 i,b,r,o,o,w,a,r,e,y,o,u,o,p,e,d,o,i,n,g,g,r,e,a,t,a,y,b,y,e,
```

jupyter Other.txt a minute ago

File Edit View Language

```
1 , , ,?, , ,!,,, ,.,
```

jupyter Upper.txt 2 minutes ago

File Edit View Language

```
1 H,H,H,O,K,
```

25. WAP that reads a text file and creates another file that is identical except that every sequence of consecutive blank spaces is replaced by a single space.

Q25

In [29]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q25():
    import os
    with open("file.txt","r") as fin:
        with open("temp.txt","w") as fout:
            l = fin.readlines()
            for i in l:
                i.replace(" ","  ")
                fout.write(i)
    os.remove("file.txt")
    os.rename("temp.txt","file.txt")
q25()
```

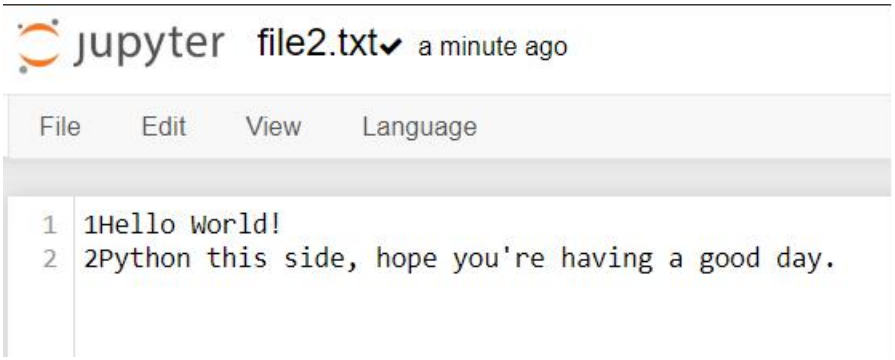
```
Date:  2020-09-11 04:51:00.056725
UserName:  jovyan
```

26 . WAP that reads a text file file1.txt and appends at the end of another file file2.txt, every line from file1.txt preceded by a line number.

Q26

In [33]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q26():
    with open('file.txt','r') as fin:
        with open('file2.txt','w') as fout:
            l = fin.readlines()
            for i in range(len(l)):
                j=i+1
                fout.write(str(j)+l[i])
q26()
```

```
Date:  2020-09-11 04:53:40.188326
UserName:  jovyan
```



Q27 WAP to find the size of file in bytes, number of words and number of lines

Q27

In [36]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q27():
    with open("file.txt","r") as fin:
        print("Number of lines", len(fin.readlines()))
        count = 0
        for i in (fin.read()).split():
            count+=1
        print('Number of words',count)
    import os
    print("Size of file in bytes= ", os.path.getsize("file.txt"))
q27()
```

```
Date:  2020-09-11 04:58:41.784908
UserName:  jovyan

Number of lines 2
Number of words 0
Size of file in bytes=  61
```

Q28. Create a binary file with name and roll number. Search for a given roll number and display the name, if not found display appropriate message

Q28

In [4]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))

with open("class_records.dat",'rb') as fin:
    while True:
        try:
            rec = pickle.load(fin)
            for i,j in list(rec.items()):
                if j == int(input("Enter Roll number to be searched")):
                    print(list(rec.items()))
                    break
        except EOFError:
            break
```

```
Date:  2020-10-13 07:35:17.798949
UserName:  jovyan
```

Q29. Create a binary file with roll number, name and marks. Input a roll number and update the marks.

Q27

In [6]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))

with open("class_records.dat",'rb') as fin:
    while True:
        try:
            pos = fin.tell()
            rec = pickle.load(fin)
            if rec['rollno']== int(input("Enter Roll no. to be updated")):
                    rec['marks'] = int(input("Enter new Marks"))
                    fin.seek(pos)
                    pickle.dump(rec, fin)

        except EOFError:
            break
print("Record Updated")
```

```
Date:  2020-10-13 07:43:40.145189
UserName:  jovyan
```

Q30 Write a menu driven telephone directory program (Binary File handling) with the following options: a) Add New Record b) Display All Records c) Search Telephone Number d) Search Name e) Update Telephone Number f) Delete a record

q30

In [3]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
import pickle
import os
print(" telephone directory program")
while True:
    print(" 1) Add New Record \n 2) Display All Records \n 3) Search Telephone Number \n 4) Search Name \n 5) Update Telephone Nu
    options=(input("Enter one of the options "))
    if options == "5":
        fin = open("Records.dat", "rb+")
        while True:
            try:
                pos = fin.tell()
                rec = pickle.load(fin)
                if rec['number']== int(input("number to be changed ")):
                    rec['number'] = int(input("New number "))
                    fin.seek(pos)
                    pickle.dump(rec, fin)

            except EOFError:
                break
        print("Record Updated")
        fin.close()

    elif options == "2":
        fin = open("Records.dat",'rb')
        while True:
            try:
                rec = pickle.load(fin)
                print(rec)
            except EOFError:
                break
        fin.close()
    elif options == "3":
        fin = open("Records.dat",'rb')
        while True:
            try:
                rec = pickle.load(fin)
                for i,j in list(rec.items()):
                    if j == int(input("Enter number to be searched ")):
                        print(list(rec.items()))
                        break
            except EOFError:
                break
        fin.close()

    elif options == "4":
        fin = open("Records.dat",'rb')
        while True:
            try:
                rec = pickle.load(fin)
                for i,j in list(rec.items()):
                    if i == (input("Enter name to be searched ")):
                        print(list(rec.items()))
                        break
            except EOFError:
                break
        fin.close()

    elif options == "1":
        d = {}
        for i in range(int(input("number of records to be added "))):
            name = input("name of the person ")
            number = int(input("The number to be added "))
            d[name]= number

        fout = open("Records.dat", 'ab')
        pickle.dump(d, fout)
        fout.close()
```

```
elif options == "1":
    d = {}
    for i in range(int(input("number of records to be added "))):
        name = input("name of the person ")
        number = int(input("The number to be added "))
        d[name]= number

    fout = open("Records.dat", 'ab')
    pickle.dump(d, fout)
    fout.close()

elif options == "6":
    fin = open("Records.dat", "rb")
    fout = open("temp.dat", "wb")
    while True:
        try:
            rec = pickle.load(fin)
            if rec['number'] == int(input("Number to be deleted ")):
                pass
            else:
                pickle.dump(rec,fout)
        except EOFError:
            break
    fin.close()
    fout.close()
    os.remove("Records.dat")
    os.rename("temp.dat","Records.dat")
    print("Record deleted")

else:
    continue
```

```
Date:  2020-10-12 17:36:16.890333
UserName:  jovyan
 telephone directory program
 1) Add New Record
 2) Display All Records
 3) Search Telephone Number
 4) Search Name
 5) Update Telephone Number
 6) Delete a record
Enter one of the options 1
number of records to be added 1
name of the person Vinayak
The number to be added 100
 1) Add New Record
 2) Display All Records
 3) Search Telephone Number
 4) Search Name
 5) Update Telephone Number
 6) Delete a record
Enter one of the options 2
{'Vinayak': 100}
 1) Add New Record
 2) Display All Records
 3) Search Telephone Number
 4) Search Name
 5) Update Telephone Number
 6) Delete a record
Enter one of the options 4
```

Q32. Considering the following definition of a dictionary MULTIPLEX, write a method in python to search and display all the content in a pickled file CINEMA.DAT, where MTYPE key of the dictionary is matching with the value 'Comedy'. MULTIPLEX = {'MNO': _____, 'MNAME': _____, 'MTYPE':_____}

Q32

In [7]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
import pickle
def search():
    with open("Cinema.dat","r") as fin:
        while True:
            try:
                rec = pickle.load(fin)
                for i,j,k in list(rec.items()):
                    if k == 'Comedy':
                        print(list(rec.items()))
                        break
            except EOFError:
                break
```

```
Date:  2020-10-12 17:43:37.124595
UserName:  jovyan
```

Q33 WAP to increase the salary by Rs 2000/- of the employee having empno as 1251 in the file emp1.dat

Q33

In [2]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
import pickle


with open('emp1.dat', 'rb+') as fin:
    while True:
        try:
            record = pickle.load(fin)
            if record["EmpNO"] == 1251:
                record["Salary"] += 2000
                fin.seek(-1, 2)
                pickle.dump(record, fin)
                break

        except EOFError:
            break
```

```
Date:  2020-10-12 19:25:39.319000
UserName:  jovyan
```

Q34 WAP to get 5 items' details (itemno, name, price, category) from the user and create a CSV file (Items.csv)

Q34

In [5]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))

import csv
with open('Items.csv', 'w', newline='') as fout:
    csv_writer = csv.writer(fout)
    csv_writer.writerow(['ItemNo.', 'Name', 'Price', 'Category'])
    for i in range(5):
        print("Item number" + str(i+1))
        itemno = input('Enter Item No ')
        name = input('Name')
        price = int(input('Price'))
        category = input('Category')
        csv_writer.writerow([itemno,name, price, category])
```

```
Date:  2020-10-12 19:35:01.117808
UserName:  jovyan
Item number1
Enter Item No IN_100
Name1K Ohm Resistor
Price88
CategoryScientific
Item number2
Enter Item No IN_101
NameGopi Dairies
Price300
CategoryBooks
Item number3
Enter Item No IN_102
NameStudy Table
Price4200
CategoryFurniture
Item number4
Enter Item No IN_103
Name9V Battery
Price260
CategoryBatteries
Item number5
Enter Item No IN_104
NameBird House
Price450
CategoryGarden and Outdoors
```

jupyter Items.csv ✔ a few seconds ago

File Edit View Language

```
1 ItemNo.,Name,Price,Category
2 IN_100,1K Ohm Resistor,88,Scientific
3 IN_101,Gopi Dairies,300,Books
4 IN_102,Study Table,4200,Furniture
5 IN_103,9V Battery,260,Batteries
6 IN_104,Bird House,450,Garden and Outdoors
7
```

Q35 . WAP to read the file Items.csv and search for an item whose itemno is obtained from the user (as input)

Q35

In [8]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
import csv
with open('Items.csv', 'r') as fin:
    csv_reader = csv.reader(fin)

    itemno = input("Search for item no")
    for row in csv_reader:
        if row[0] == itemno:
            print(row)
            break
```

```
Date:  2020-10-12 19:41:39.715383
UserName:  jovyan
Search for item noIN_104
['IN_104', 'Bird House', '450', 'Garden and Outdoors']
```

Q36. WAP to read the file Items.csv and create a file highitems.csv, containing only those item details from Items.csv where price > 250.

Q36

In [28]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
import csv
with open('Items.csv', 'r') as fin:
    with open("highitems.csv", 'w') as fout:
        csv_reader = csv.reader(fin)
        csv_writer = csv.writer(fout)

        for i in csv_reader:
            try:
                if int(i[2]) > 250:
                    csv_writer.writerow(i)
            except:
                continue
```

```
Date:  2020-10-12 19:56:39.517796
UserName:  jovyan
```

jupyter highitems.csv a minute ago

File Edit View Language

```
1 IN_101,Gopi Dairies,300,Books
2 IN_102,Study Table,4200,Furniture
3 IN_103,9V Battery,260,Batteries
4 IN_104,Bird House,450,Garden and Outdoors
5
```

Q37 WAP to create the same csv file as Q34 but with a delimiter character as pipe (|)

Q37

In [29]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))

import csv
with open('Items.csv', 'w', newline='') as fout:
    csv_writer = csv.writer(fout, delimiter='|')
    csv_writer.writerow(['ItemNo.', 'Name', 'Price', 'Category'])
    for i in range(5):
        print("Item number" + str(i+1))
        itemno = input('Enter Item No ')
        name = input('Name')
        price = int(input('Price'))
        category = input('Category')
        csv_writer.writerow([itemno,name, price, category])
```

```
Date:  2020-10-12 20:00:11.507490
UserName:  jovyan
Item number1
Enter Item No IN_100
Name1K Ohm Resistor
Price88
CategoryScientific
Item number2
Enter Item No IN_101
NameGopi Dairies
Price300
CategoryBooks
Item number3
Enter Item No IN_102
NameStudy Table
Price4200
CategoryFurniture
Item number4
Enter Item No IN_103
Name9V Battery
Price260
CategoryBatteries
Item number5
Enter Item No IN_104
NameBird House
Price450
CategoryGarden and Outdoors
```

jupyter Items.csv ✔ a minute ago

File Edit View Language

```
ItemNo.|Name|Price|Category
IN_100|1K Ohm Resistor|88|Scientific
IN_101|Gopi Dairies|300|Books
IN_102|Study Table|4200|Furniture
IN_103|9V Battery|260|Batteries
IN_104|Bird House|450|Garden and Outdoors
```

Q.38 Create a 2D list in Python that stores runs scored by a batsman in five overs: Runs = [[0, 6, 4, 1, 0, 0], [3, 0, 2, 0, 0, 0], [0, 0, 4, 4, 0, 1], [0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 0]]

Q38

In [10]:

```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q38():
    Runs = [[0, 6, 4, 1, 0, 0], [3, 0, 2, 0, 0, 0], [0, 0, 4, 4, 0, 1], [0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 0]]
    count = over = total= 0
    l = []
    for i in Runs:
        temp = 0
        for j in i:
            temp +=j
            total +=j
        l.append(temp)
        if temp > count:
            count,over = temp, Runs.index(i)+1
    l.sort()
    print("Highest score was",count,'in the over ',over)
    print("Minimun runs",l[0])
    print("TOTAL score was ", total )
q38()
```

```
Date:  2020-09-11 06:07:37.999478
UserName:  jovyan

Highest score was 11 in the over  1
Minimun runs 0
TOTAL score was  26
```

Q39. Write definition of a method AFIND(CITIES) to display all the city names from a list of cities, which are starting with alphabet 'A'

Q39

In [4]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q39():
    def afind(arr):
        for i in arr:
            if i[0] == 'A':
                print(i)
    afind(['Ahemdabad','Agra','Alipura','Dhaka','Delhi','Zila'])

q39()
```

```
Date:  2020-09-11 05:44:22.372810
UserName:  jovyan

Ahemdabad
Agra
Alipura
```

Q40 Write a menu driven program to implement a stack for the students (adm_no, name).

Q40

In [13]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q40():
    maxlen = 5
    stack= []
    print("Menu")
    print(" 1.Insert Element \n 2.Delete Element \n 3.Display Stack \n 4.Exit")

    while True:
        option = int(input("Enter one of the options  "))
        if option == 1:
            if len(stack)== maxlen:
                print("Overflow")
            else:
                num = int(input("Enter admission number  "))
                name  = input("Enter name of the student ")
                stack.append((num,name))
        elif option ==2:
            if stack == []:
                print("Underflow")
            else:
                print("Deleted the element  ", stack.pop())
        elif option ==3 :
            print(stack)
        elif option == 4:
            break
        else:
            print("Please enter a valid input ")
            continue
q40()
```

```
Date:  2020-09-11 06:18:23.274952
UserName:  jovyan

Menu
 1.Insert Element
 2.Delete Element
 3.Display Stack
 4.Exit
Enter one of the options  1
Enter admission number  01
Enter name of the student Vinayak Bector
Enter one of the options  1
Enter admission number  02
Enter name of the student Meer Sisodia
Enter one of the options  3
[(1, 'Vinayak Bector'), (2, 'Meer Sisodia ')]
Enter one of the options  2
Deleted the element   (2, 'Meer Sisodia ')
Enter one of the options  2
Deleted the element   (1, 'Vinayak Bector')
Enter one of the options  3
[]
Enter one of the options  4
```

Q42. Write a function in python, MakePush(Package) and MakePop(Package) to add a new Package and delete a Package from a List of Package Description, considering them to act as push and pop operations of the Stack data structure.

Q41

In [16]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q41():
    l = input("Enter your list separated by commas").split()
    print('Your list',l)
    def insertq(arr,data):
        arr.append(data)
    def deleteq(arr):
        if arr == []:
            print("Empty list")
        else:
            return arr.pop()
    data = input("Enter data that needs to be added ")
    insertq(l,data)
    print('Updated list',l)
    print('Deleting data')
    print(deleteq(l))
q41()
```

```
Date:  2020-09-11 06:28:07.954401
UserName:  jovyan

Enter your list separated by commascats,dogs,animals
Your list ['cats,dogs,animals']
Enter data that needs to be added jack
Updated list ['cats,dogs,animals', 'jack']
Deleting data
jack
```

Q42

In [5]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q42():
    stack = []
    def Make_Package(pack):
        stack.append(pack)

    def Make_Pop():
        if stack == []:
            return ("No elements in the stack ")
        else:
            return stack.pop()
    def Display():
        return stack
    #Deleting to test underflow
    print(Make_Pop())
    #Adding three packs
    Make_Package("Bro")
    Make_Package("Code")
    Make_Package("Jones")
    #Displaying the stack
    print(Display())
    #Deleting pack
    print(Make_Pop())
    #Displaying the stack
    print(Display())
q42()
```

```
Date:  2020-09-11 06:45:15.273530
UserName:  jovyan

No elements in the stack
['Bro', 'Code', 'Jones']
Jones
['Bro', 'Code']
```

Q43 WAP to print a string in reverse order. (Hint: Extract individual characters from string and push in stack; once done; keep popping from stack)

q43

In [8]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q43():
    string  = str(input("Enter your string "))
    stack  = []
    for i in string:
        stack.append(i)
    print("Your orginal string", string)
    for i in range(len(string)):
        print(stack.pop(-1),end='')
q43()
```

```
Date:  2020-09-11 05:55:21.675905
UserName:  jovyan

Enter your string Vinayak Bector
Your orginal string Vinayak Bector
rotceB kayaniV
```

Q44 WAP to implement a queue of order numbers in a restaurant.

Q44

In [6]:
```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q44():
    import random
    print("Yummy programme")
    print("1.Order a meal \n 2.Waiting Queue \n 3.Order is Ready \n Exit")
    q  = []
    while True:
        x = int(input("Enter any one of the above options "))
        if x == 1:
            a = random.randint(100,999)
            print("Your order number is", a)
            q.append(a)
        elif x ==2:
            print("The order list", q)
        elif x == 3:
            if q==[]:
                print("No orders in queue")
                continue
            else:
                a = q.pop(0)
                print("Order is ready ", a )
        elif x ==4:
            break
        else:
            print("Please give a valid input")
            continue
q44()
print("EOF")
```

```
Date:  2020-09-11 05:49:58.692946
UserName:  jovyan

Yummy programme
1.Order a meal
 2.Waiting Queue
 3.Order is Ready
 Exit
Enter any one of the above options 1
Your order number is 309
Enter any one of the above options 1
Your order number is 145
Enter any one of the above options 1
Your order number is 734
Enter any one of the above options 1
Your order number is 490
Enter any one of the above options 2
The order list [309, 145, 734, 490]
Enter any one of the above options 3
Order is ready  309
Enter any one of the above options 3
Order is ready  145
Enter any one of the above options 1
Your order number is 201
Enter any one of the above options 2
The order list [734, 490, 201]
Enter any one of the above options 5
Please give a valid input
Enter any one of the above options 4
EOF
```

Q45 WAP to implement a stack of URLs in a web browser.

Q45

In [9]:

```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
print()
def q45():
    print("URL HISTORY")
    print(" 1.Go to new URL \n 2.History \n 3.Go Back \n 4.Exit")
    stack  = []
    while True:
        x = int(input("Enter any one of the above options "))
        if x == 1:
            print("New URL")
            a = input("Enter your URL")
            stack.append(a)
        elif x ==2:
            print("HISTORY", stack)
        elif x == 3:
            if stack == []:
                print("No URLs Visited till now")
            else:
                a = stack.pop(-1)
                print("Last visited ", a )
        elif x ==4:
            break
        else:
            print("Please give a valid input")
            continue
q45()
print("EOF")
```

```
Date:  2020-09-11 05:57:24.005598
UserName:  jovyan

URL HISTORY
 1.Go to new URL
 2.History
 3.Go Back
 4.Exit
Enter any one of the above options 1
New URL
Enter your URLgoogle
Enter any one of the above options 1
New URL
Enter your URLYouTube
Enter any one of the above options 1
New URL
Enter your URLspvdelhi
Enter any one of the above options 2
HISTORY ['google', 'YouTube', 'spvdelhi']
Enter any one of the above options 3
Last visited  spvdelhi
Enter any one of the above options 3
Last visited  YouTube
Enter any one of the above options 3
Last visited  google
Enter any one of the above options 1
New URL
Enter your URLCNN
Enter any one of the above options 2
HISTORY ['CNN']
Enter any one of the above options 4
EOF
```