PYTHON SEE LAB SOLUTIONS by the_back_bencher_3

Note: ”I have prepared the solutions to the best of my knowledge

Points to be kept in mind before studying

- **Please don’t go to mug up,, It is beneficial if these programs are executed in eclipse, So that you can improve your error correcting skill and will help during exams and it’ll also give u the idea about the code**
- **If there are any mistakes, do correct it by yourself, I’ m not responsible**
  If you don’t get output
  ”

Part A

1. The finance department of a company wants to calculate the monthly pay of one of its employee. Monthly pay should be calculated as mentioned in the formula below and display all the employee details. Monthly Pay= No. of hours worked in a week * Pay rate per hour * No .of weeks in a
   Month. Write a Python Program to implement the problem

```
Created on 11-Nov-2018

@author: Terri
'''
week_hours = float(input("Enter weekly working hours : "))
hourly_pay = float(input("Enter pay rate per hour : "))
working_weeks = float(input("Enter working weeks in a month : "))
monthly_pay = week_hours * hourly_pay * working_weeks
print("Calculated monthly pay : Rs", monthly_pay)

"""
Enter weekly working hours : 25
Enter pay rate per hour : 1000
Enter working weeks in a month : 4
Calculated monthly pay : Rs 100000.0
"""
```

2.

The finance department wants to calculate the monthly net pay of one of its employee by finding the income tax to be paid and the net salary after the income tax deduction. The employee should pay the income tax based on the following table: Display basic salary, allowances, gross pay, income tax and net pay

Gross Salary	**Tax Percentage**
Below 5,000	Nil
5,001 to 10,000	10%
10,001 to 20,000	20%
More than 20,000	30%

```
'''
Created on 11-Nov-2018

@author: Terri
'''
emp_id = int(input("Enter Employee ID : "))
basic_pay = float(input("Enter Basic Pay : "))
allowances = float(input("Enter Allowances : "))
gross_pay = basic_pay + allowances
if gross_pay > 20000:
    income_tax = gross_pay * 0.3
elif gross_pay > 10000:
    income_tax = gross_pay * 0.2
elif gross_pay > 5000:
    income_tax = gross_pay * 0.1
else:
    income_tax = 0
net_pay = gross_pay - income_tax
print("Employee ID : ", emp_id)
print("Basic Pay : Rs", basic_pay)
print("Allowances : Rs", allowances)
print("Gross Pay : Rs", gross_pay)
print("Income Tax : Rs", income_tax)
print("Net Pay : Rs", net_pay)

"""output
```

```
Enter Employee ID : 101
Enter Basic Pay : 25000
Enter Allowances : 4500
Employee ID :  101
Basic Pay : Rs 25000.0
Allowances : Rs 4500.0
Gross Pay : Rs 29500.0
Income Tax : Rs 8850.0
Net Pay : Rs 20650.0
      """
```

3.

Write a Python program to generate first ‘n ’Fibonacci numbers.

```
f1=0
f2=1
n=int(input("Enter the number of fibonacci numbers to be printed"))
if n==0:
    print("invalid input")
if n==1:
    print(f1)
if n>=2:
    print(f1)
    print(f2)
    if n>2:
        for i in range(2,n):
            f3=f1+f2
            print(f3)
            f1=f2
            f2=f3



            """
            output
              Enter the number of fibonacci numbers to be printed5
0
1
1
2
2
```

4.

Given below is the list of marks scored by students. Find top three scorers for the course and also display the average marks scored by all students. Implement the solution using Python Programming.

Student Name	Marks
John	86.5
Jack	91.2
Jill	84.5
Harry	72.1
Joe	80.5

```
'''
Created on 11-Nov-2018

@author: Terri
'''
marks_list = { 'John': 86.5, 'Jack': 91.2, 'Jill': 84.5, 'Harry': 72.1, 'Joe':
80.5 }
sorted_keys = sorted(marks_list, key=marks_list.get, reverse=True)
print("Top 3 students: ")
for student in sorted_keys[:3]:
    print(student, " : ", marks_list[student])
sum = 0
for student in marks_list:
    sum += marks_list[student]

"""
output

Top 3 students:
Jack  :  91.2
John  :  86.5
Jill  :  84.5
     """
```

Note: Sorting any sequence is very easy in Python using built-in method sorted() which does all the hard work for you.
sorted () sorts any sequence (list, tuple) and always returns a list with the elements in sorted manner, without modifying the original sequence.
Syntax: sorted (iterable, key, reverse)

Parameters: sorted takes three parameters from which two are optional.

· *Iterable :* sequence (list, tuple, string) or collection (dictionary, set, frozenset) or any other iterator that needs to be sorted.

· *Key*(optional) : A function that would server as a key or a basis of sort comparison.

· *Reverse*(optional) : If set true, then the iterable would be sorted in reverse (descending) order, by default it is set as false.

7.

5. Write a program to count the number of capital letters and display the position of each capital letter in a user entered string via keyboard

```
'''
Created on 10-Nov-2018

@author: Terri
'''
str1=input("Enter the String ");
lis=[]
lis2=[]
for i in range(0,len(str1)):
    if str1[i].isupper():
        lis.append(str1[i])
        lis2.append(i)
print("Number of capital letters--",len(lis))
print("Capital letter       Position")
for i in range(0,len(lis)):
    print(lis[i],"                  ",lis2[i])

"""
 output
 Enter the String Joel NazaREth
Number of capital letters-- 4
Capital letter       Position
J                    0
N                    5
R                    9
E                    10

"""
```

6. Write a program to count the number of each vowel in a given string

```
#vowel count
str1=input("Enter the string")
str2=str1.lower()
```

```
count=0
dict={'a':0,'e':0,'i':0,'o':0,'u':0}
for i in range(0,len(str1)):
    if(str2[i]=='a' or str2[i]=='e' or str2[i]=='i' or str2[i]=='o' or
str2[i]=='u'):
        dict[str2[i]]+=1
        count+=1
print(dict)
print("Number of vowels in the string is ",count)


"""output
Enter the stringjoel john Nazareth
{'a': 2, 'e': 2, 'i': 0, 'o': 2, 'u': 0}
Number of vowels in the string is  6

"""
```

7. Write a program to remove all punctuations like “'!()-[]{};:’’’,\,<>,/,?,@,#,$,%^&*_~” from the string provided by the user.

```
'''
Created on 11-Nov-2018

@author: Terri
'''
def remove_punctuations(string):
    res =""
    for char in string:
        if char not in "'!()-[]{};:' '',\<>/?@#$%&^&*_~":
            res+= char
    return res
string = input("Enter a string : ")
print("your string without punctuations : ", remove_punctuations(string))

"""
 output

 Enter a string : df%^%lkjghl@!!lkfj&(807-56$#$
your string without punctuations :  dflkjghllkfj80756
     """
```

8. Consider two strings, String1 and String2 and display the merged string as output. The merged string should be the capital letters from both the strings in the order they appear.
Sample Input:String1:ILikeC String2:MaryLikesPython
Merged string should be ILCMLPS

```
'''
Created on 19-Sep-2018

@author: Terri
'''
#merge capital letters of two string
str1=input("Enter the string 1")
str2=input("Enter the string 2")
str3=""
for i in range(0,len(str1)):
    if str1[i].isupper():
        str3+=str1[i]

for i in range(0,len(str2)):
    if str2[i].isupper():
        str3+=str2[i]
print("Merged string is ",str3)

"""
output
Enter the string 1JohnNy
Enter the string 2SinS
Merged string is  JNSS”””
```

9. 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.

```
'''
Created on 19-Sep-2018

@author: Terri
'''#binary search  ,,list of items should be in sorted order
list1=[]
n=int(input("Enter the number of items"))
for i in range(0,n):
    list1.append(int(input("Enter the integer")))
list1.sort();
print(list1)


low=0
high=n-1
flag=0
key=int(input("Enter the search element"))
while low<=high:
    mid=int((low+high)/2)
    if list1[mid]==key:
        print("Key found at the position ",mid+1)
```

```
        flag=1
        break
    elif list1[mid]>key:
        high=mid-1
    else:
        low=mid+1
if flag==0:
    print(key," not found")

"""
output

Enter the number of items5
Enter the integer65
Enter the integer78
Enter the integer9
Enter the integer7
Enter the integer43
[7, 9, 43, 65, 78]
Enter the search element65
Key found at the position  4
     """
```

10. Write a function that returns the index of the smallest element in a list of integers. If the number of such elements is greater than 1,return the smallest index. Use the following header:
def indexOfSmallestElement(lst):

Write a test program that prompts the user to enter a list of numbers, invokes this function to return the index of the smallest element and displays the index.

```
'''
Created on 19-Sep-2018

@author: Terri
'''
#index of smallest integer in the list
def indexOfSmallestElement(lst):
    t=min(lst)
    return lst.index(t)


list1=[]
n=int(input("Enter the number of elements"))
for i in range(0,n):
    list1.append(int(input("Enter the integer")))
print("The index of the smallest element ",indexOfSmallestElement(list1))
```

```
"""
 output
 Enter the number of elements7
Enter the integer65
Enter the integer8
Enter the integer7
Enter the integer56
Enter the integer12
Enter the integer4
Enter the integer714
      The index of the smallest element  5 """
```

11. Write a program that will count the number of characters, words,and lines in a file. Words are separated by a white-space character. Your program should prompt the user to enter a filename.

```
'''
Created on 04-Sep-2018

@author: Terri
''
'''


fo=open("file.txt","w+")
str1=input("ENTER THE TEXT")
str2=" is the best"
fo.write(str1)
fo.write(str2)
fo.close()
#file to be opened should be created already as i have created in the program
itself
#file can be opened using open or with method
#fo=open("file.txt","r+")
str2=input("Enter the filename")
with open(str2,'r') as fo:
    nl=nw=nc=0

    for line in fo:
        nl=nl+1
        words=line.split()
        #print(words)
        nw=nw+len(words)
        for i in words:
            nc=nc+len(i)
```

```
print("Number of lines ",nl,"\nNumber of words ",nw,"\nNumber of characters ",nc)

file.txt contains
cse5D is the best

"""Output
ENTER THE TEXTcse5d
Enter the filenamefile.txt
Number of lines  1
Number of words  4
Number of characters  14
"""
```

12. Suppose that a text file contains marks for 6 courses for a student in a line. Each course marks is separated by space as delimiter. File contains marks for 'n' number of students in separate lines. Write a program that reads the marks from the file for each student and displays the total and average. Your program should prompt the user to enter a filename.

```
'''
Created on 11-Nov-2018

@author: Terri
'''
file_name = input("Enter file name : ")
try:
    file = open(file_name)
    scores =0
    for line in file:
        line = line.split(' ')
        print(line)
        for item in range(len(line)) :
            scores += int(line[item])
        print("Total score : ",scores)
        print("Average score : ", (scores/len(line)))
        scores=0
except FileNotFoundError:
    print("file not found")


file2.txt contains
15 12 13 16 12 19
13 14 17 18 21 33
"""Output

Enter file name : file2.txt
```

```
['15', '12', '13', '16', '12', '19\n']
Total score :  87
Average score :  14.5
['13', '14', '17', '18', '21', '33']
Total score :  116
Average score :  19.333333333333332"""
```

Part-B

13. Design a class named Rectangle to represent a rectangle. class contains:
Two data fields named width and height.
A constructor that creates a rectangle with the specified width and height.
A method named getArea() that returns the area of this rectangle. Write a program that creates 'n' number of Rectangle objects. Read values of width and height from the key board for each Rectangle instance. Display the width and height of the rectangle having maximum and min area

```
'''
Created on 11-Nov-2018

@author: Terri
'''
class rectangle:
    def __init__(self,l,b):
        self.l=l
        self.b=b
    def getArea(self):
        return (self.l*self.b)

n=int(input("Enter the number of Rectangle objects "))
lis=[]
lis1=[]
for i in range(n):
    w=float(input("Enter the width of rectangle %d"%(i+1)))
    h=float(input("Enter the height of rectangle %d"%(i+1)))
    lis.append(rectangle(w,h))
    lis1.append(lis[i].getArea())

i_of_max=lis1.index(max(lis1))
i_of_min=lis1.index(min(lis1))
print(max(lis1))
print("The width and height of the rectangle having max area ",lis[i_of_max].l,"
",lis[i_of_max].b)
print("The width and height of the rectangle having max area ",lis[i_of_min].l,"
",lis[i_of_min].b)
```

```
    """
    output
    Enter the number of Rectangle objects 2
Enter the width of rectangle 15
Enter the height of rectangle 18.2
Enter the width of rectangle 214
Enter the height of rectangle 25
70.0
The width and height of the rectangle having max area  14.0   5.0
The width and height of the rectangle having max area  5.0   8.2
"""
```

14. Design a class named Account that contains:
A private int data field named accountno for the account.
A private float data field named balance for the account.
A constructor that creates an account with the specified accountno and
Initial balance (default 100).
A method named withdraws that withdraws a specified amount from the account. A minimum balance of 100 should be maintained for each account.
A method named deposit that deposits a specified amount to the specific account.
Write a program that maintains 'n' number of Account objects with unique accountno and supports following operations. a) New account creation b) deposit operation for a given accountno c) withdraw operation for a given account no d) Display account no with highest balance

```
'''
Created on 11-Nov-2018

@author: Terri
'''

class Account:
    def __init__(self,accno):
        self.__accno=accno
        self.__balance=100.0
    def deposit(self,amt):
        self.__balance+=amt
    def withdraw(self,amt):
        l=self.__balance
        if((l-amt)<100):
            print("insufficient balance")
        else:
            self.__balance-=amt
            print("Successfully witthdrawn")
    def getbalance(self):
        return self.__balance
#n=int("Enter the number of accounts"
```

```
lis=[]
lis2=[] #maintains accno
lis3=[]# maintains balance
f=True
while(f):
    print("1.New Account creation\n2.Deposit \n3.Withdraw\n4.Display accno having
highest balance\n5.Exit")
    n=int(input("Enter the choice"))
    if(n==1):
        accno=int(input("Enter the accno"))
        if(accno in lis2):
            print("Account number already exists")
        else:
            lis.append(Account(accno))
            lis2.append(accno)
    elif(n==2):
        accno=int(input("Enter the existing accno"))
        try:
            i=lis2.index(accno)
            amt=float(input("Enter the amount to be deposited"))
            lis[i].deposit(amt)
            print("Successfully deposited")
        except:
            print("accno doesnt exists")
    elif(n==3):
        accno=int(input("Enter the existing accno"))
        try:
            i=lis2.index(accno)
            amt=float(input("Enter the amount to be withdrawn"))
            lis[i].withdraw(amt)
        except:
            print("accno doesnt exists")
    elif(n==4):
        for i in range(len(lis)):
            lis3.append(lis[i].getbalance())

        index1=lis3.index(max(lis3))
        print("The account no having maximum balance ",lis2[index1])
    elif(n==5):
        f=False
    else:
        print("Invalid choice")

    print()
```

```
"""
output

1.New Account creation
2.Deposit
3.Withdraw
4.Display accno having highest balance
5.Exit
Enter the choice1
Enter the accno101

1.New Account creation
2.Deposit
3.Withdraw
4.Display accno having highest balance
5.Exit
Enter the choice1
Enter the accno101
Account number already exists

1.New Account creation
2.Deposit
3.Withdraw
4.Display accno having highest balance
5.Exit
Enter the choice1
Enter the accno102

1.New Account creation
2.Deposit
3.Withdraw
4.Display accno having highest balance
5.Exit
Enter the choice2
Enter the existing accno101
Enter the amount to be deposited258
Successfully deposited

1.New Account creation
2.Deposit
3.Withdraw
4.Display accno having highest balance
5.Exit
Enter the choice3
Enter the existing accno102
```

```
Enter the amount to be withdrawn100
insufficient balance

1.New Account creation
2.Deposit
3.Withdraw
4.Display accno having highest balance
5.Exit
Enter the choice4
The account no having maximum balance  101

1.New Account creation
2.Deposit
3.Withdraw
4.Display accno having highest balance
5.Exit
Enter the choice5

"""
```

15. Develop python program to perform the below mentioned operations.

a) display total marks scored by each student

b) Display top scorer and the top score

Scenario: There are 8 students and answers to 10 multiple choice questions of each student is stored in a file called **marks.txt**. Each answer is delimited by space. Each line provides a student's answers to the questions, as shown below. The answer key is

stored in a file named ***keys.txt***. The format of answer keys as shown below.

Key D B D C C D A E A D

Marks.txt

Student0 A B A C C D E E A D

Student1 D B A B C A E E A D

Student2 E D D A C B E E A D

Student3 C B A E D C E E A D

Student4 A B D C C D E E A D

B B E C C D E E A D
Student5
B B A C C D E E A D
Student6
E B E C C D E E A D
Student7

```
'''
Created on 11-Nov-2018

@author: Terri
'''
ans=open("KEYS.txt","r").readline().split()
f=open("MARKS.txt","r")
for line in f:
    t=line.split()
    nm=t[0]
    count=0
    for x in range(1,len(ans)+1):
        if t[x]==ans[x-1]:
            count+=1
    print(nm," Marks =",count,"/ 10")

"""Note: create two text files KEYS.txt and MARKS.txt
KEYS.txt (contents)
D B D C C D A E A D
MARKS.txt (contents)
Student0 A B A C C D E E A D
Student1 D B D C C D A E A D
Student2 E D D A C B E E A D
Student3 C B A E D C E E A D
Student4 A B D C C D E E A D
Student5 B B E C C D E E A D
Student6 B B A C C D E E A D
Student7 E B E C C D E E A D


 output

 Student0  Marks = 7 / 10
Student1  Marks = 10 / 10
Student2  Marks = 5 / 10
```

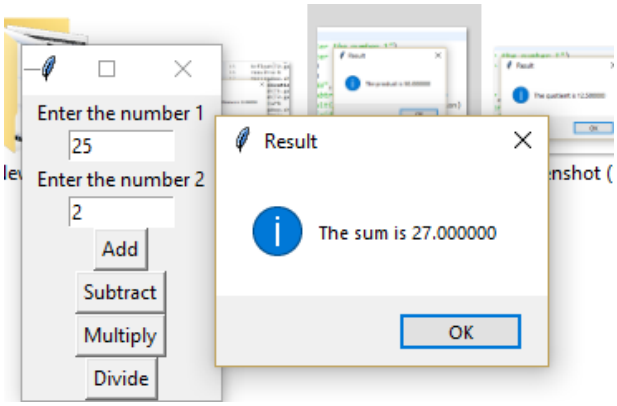
```
Student3  Marks = 4 / 10
Student4  Marks = 8 / 10
Student5  Marks = 7 / 10
Student6  Marks = 7 / 10
Student7  Marks = 7 / 10
"""
```

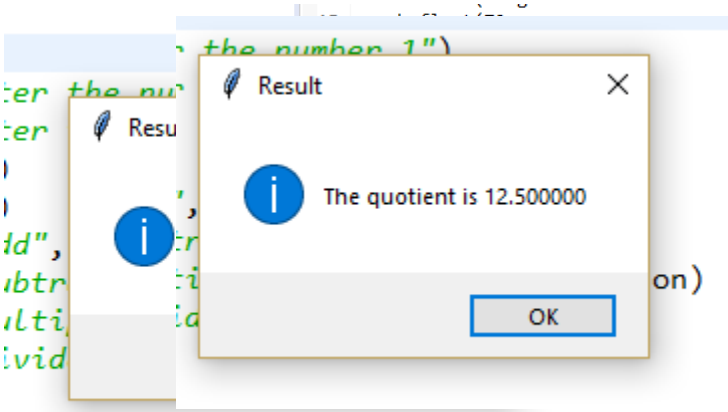
16. Develop a Tkinter based python program to perform arithmetic operations add,subtract.mul and div of two numbers. The numbers should be read using two different text fields and there should be four separate buttons for performing each operation. On clicking the button, the result should be displayed in a separate message dialog box

```
'''
Created on 11-Nov-2018

@author: Terri
'''
from tkinter import *
from tkinter import messagebox
def addition():
    a=float(T1.get())
    b=float(T2.get())
    sum1=a+b
    messagebox.showinfo("Result","The sum is %f"%(sum1))
def subtraction():
    a=float(T1.get())
    b=float(T2.get())
    result=a-b
    messagebox.showinfo("Result","The difference is %f"%(result))
def multiplication():
    a=float(T1.get())
    b=float(T2.get())
    result=a*b
    messagebox.showinfo("Result","The product is %f"%(result))
def division():
    a=float(T1.get())
    b=float(T2.get())
    sum1=a/b
    messagebox.showinfo("Result","The quotient is %f"%(sum1))
window=Tk()

L1=Label(window,text="Enter the number 1")
L2=Label(window,text="Enter the number 2")
T1=Entry(window,width=10)
T2=Entry(window,width=10)
B1=Button(window,text="Add",command=addition)
B2=Button(window,text="Subtract",command=subtraction)
B3=Button(window,text="Multiply",command=multiplication)
```

```
B4=Button(window,text="Divide",command=division)
L1.pack()
T1.pack()
L2.pack()
T2.pack()
B1.pack()
B2.pack()
B3.pack()
B4.pack()
window.mainloop()
```

output





17. Write a program to create a list to maintain country names, respective capital and its population. The program should support following operations
a) To enter country name, capital and its population.
b) To accept the name of a country as an input and print the corresponding capital name and population as output. Otherwise, the program should print an appropriate message if the country is not found in the list.
c) To display the country name with highest population.

```
'''
Created on 11-Nov-2018

@author: Terri
'''
class Country:
    def __init__(self,name,capital,population):
        self.name=name
        self.cap=capital
        self.pop=population
```

```
    def printDetails(self):
        print("Country:",self.name," Capital:",self.cap," Population:",self.pop)
lis=[]
while True:
    print("*"*10)
    ch=int(input("1.Add 2.Display by name 3.Display highest population
4.Exit\nEnter option:"))

    if ch==4:break
    if ch==1:
        n=input("enter country name\n")
        c=input("enter capital\n")
        p= int(input("enter population\n"))
        cc=Country(n,c,p)
        lis.append(cc)
    if ch==2:
        name=input("Enter name of country:")
        found=False
        for x in lis:
            if x.name == name:
                x.printDetails()
                found=True
                break
        if found==True:
            continue
        else:
            print("Country name not found...")
    if ch==3:
        found=False
        if len(lis)==0:
            print("Empty list...")
            continue
        high=lis[0]
        for x in lis:
            if x.pop>high.pop:
                high=x
        print("Highest Population:")
        for x in lis:
            if x.pop==high.pop:
                found=True
                x.printDetails()
        if found==False:
            print("None")


"""
```

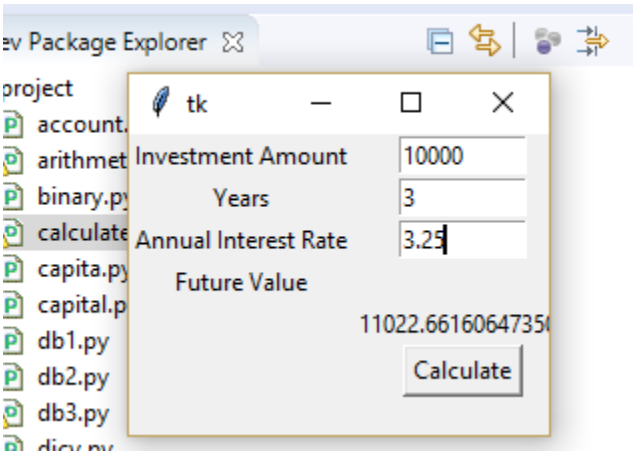
```
output_____


**********
1.Add 2.Display by name 3.Display highest population 4.Exit
Enter option:1
enter country name
India
enter capital
Delhi
enter population
1200000
**********
1.Add 2.Display by name 3.Display highest population 4.Exit
Enter option:1
enter country name
China
enter capital
Beijing
enter population
1500000
**********
1.Add 2.Display by name 3.Display highest population 4.Exit
Enter option:2
Enter name of country:India
Country: India  Capital: Delhi  Population: 1200000
**********
1.Add 2.Display by name 3.Display highest population 4.Exit
Enter option:3
Highest Population:
Country: China  Capital: Beijing  Population: 1500000
**********
1.Add 2.Display by name 3.Display highest population 4.Exit
Enter option:4""""
```

18. Write a program that calculates the future value of an investment at a given interest rate for a specified number of years. The formula for the calculation is as follows:
futureValue=investmentAmount*(1+monthlyInterestRate)years*12
Use textfields for users to enter the investment amount,years,and interest rate. Display the future amount in a label field when the user clicks the Calculate button, as shown in Figure above.

```
'''
Created on 11-Nov-2018

@author: Terri
'''
```

```
from tkinter import *
def clicked():
    a=float(t1.get())
    b=float(t2.get())
    c=float(t3.get())
    mr=(c/100)/12 #computing monthly interest
    print(a)
    print(b)
    print(c)
    fv=a*pow((1+mr),b*12)
    l5.configure(text=fv)

window=Tk()
window.geometry("200x150")
l1=Label(window,text="Investment Amount")
l2=Label(window,text="Years")
l3=Label(window,text="Annual Interest Rate")
l4=Label(window,text="Future Value")
l5=Label(window)
t1=Entry(window,width=10)
t2=Entry(window,width=10)
t3=Entry(window,width=10)
b=Button(window,text="Calculate",command=clicked)
l1.grid(column=0,row=0)
l2.grid(column=0,row=1)
l3.grid(column=0,row=2)
l4.grid(column=0,row=3)
l5.grid(column=1,row=4)
t1.grid(column=1,row=0)
t2.grid(column=1,row=1)
t3.grid(column=1,row=2)
b.grid(column=1,row=5)
window.mainloop()
```

output

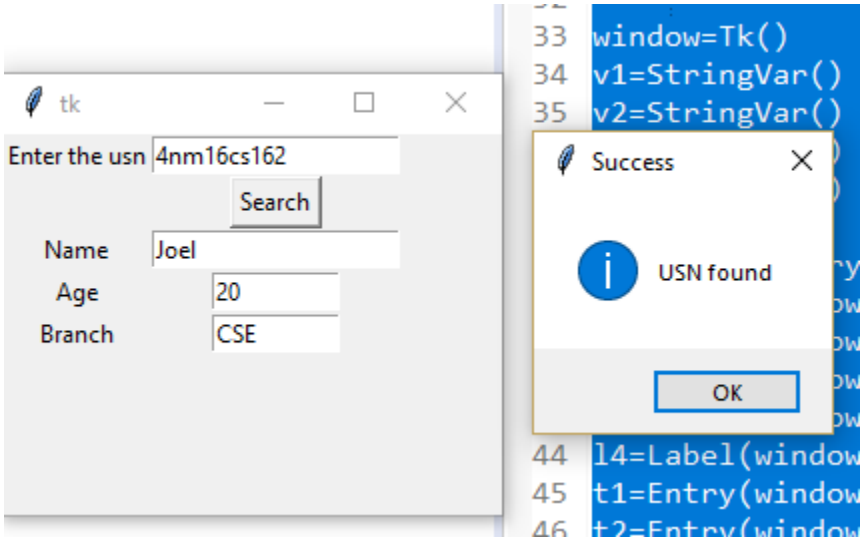


19. Using Tkinter interface display student details such as **name(textfield),age(textfield),branch(textfield)** by performing search operation for the user entered usn via textfield. Information is displayed upon click on the search button. Students information is stored in a database by considering the table Student(USN:String,Name:String,Age:Int,Branch:String).Display the success and failure message using MessageBox.

```
'''
Created on 11-Nov-2018

@author: Terri
'''
from tkinter import *;
from tkinter import messagebox
import pymysql

def search():
    db = pymysql.connect("localhost","root","","college" )
    cursor = db.cursor()
    sql="select * from student where usn='%s'"%(v1.get())
    try:
        cursor.execute(sql)
        data=cursor.fetchall()
        for row in data:
            name=row[1]
            age=row[2]
            branch=row[3]
        v2.set(name)
        v3.set(age)
        v4.set(branch)
        messagebox.showinfo("Success","USN found")
    except:
        messagebox.showerror("Error","Unsuccessful")
        db.close()




window=Tk()
```

```
v1=StringVar()
v2=StringVar()
v3=StringVar()
v4=StringVar()

window.geometry("250x190")
l1=Label(window,text="Enter the usn")
b=Button(window,text="Search",command=search)
l2=Label(window,text="Name")
l3=Label(window,text="Age")
l4=Label(window,text="Branch")
t1=Entry(window,width=20,textvariable=v1)
t2=Entry(window,width=20,textvariable=v2)
t3=Entry(window,width=10,textvariable=v3)
t4=Entry(window,width=10,textvariable=v4)
l1.grid(column=0,row=0)
l2.grid(column=0,row=2)
l3.grid(column=0,row=3)
l4.grid(column=0,row=4)

t1.grid(column=1,row=0)
t2.grid(column=1,row=2)
t3.grid(column=1,row=3)
t4.grid(column=1,row=4)
b.grid(column=1,row=1)
window.mainloop()
```

output



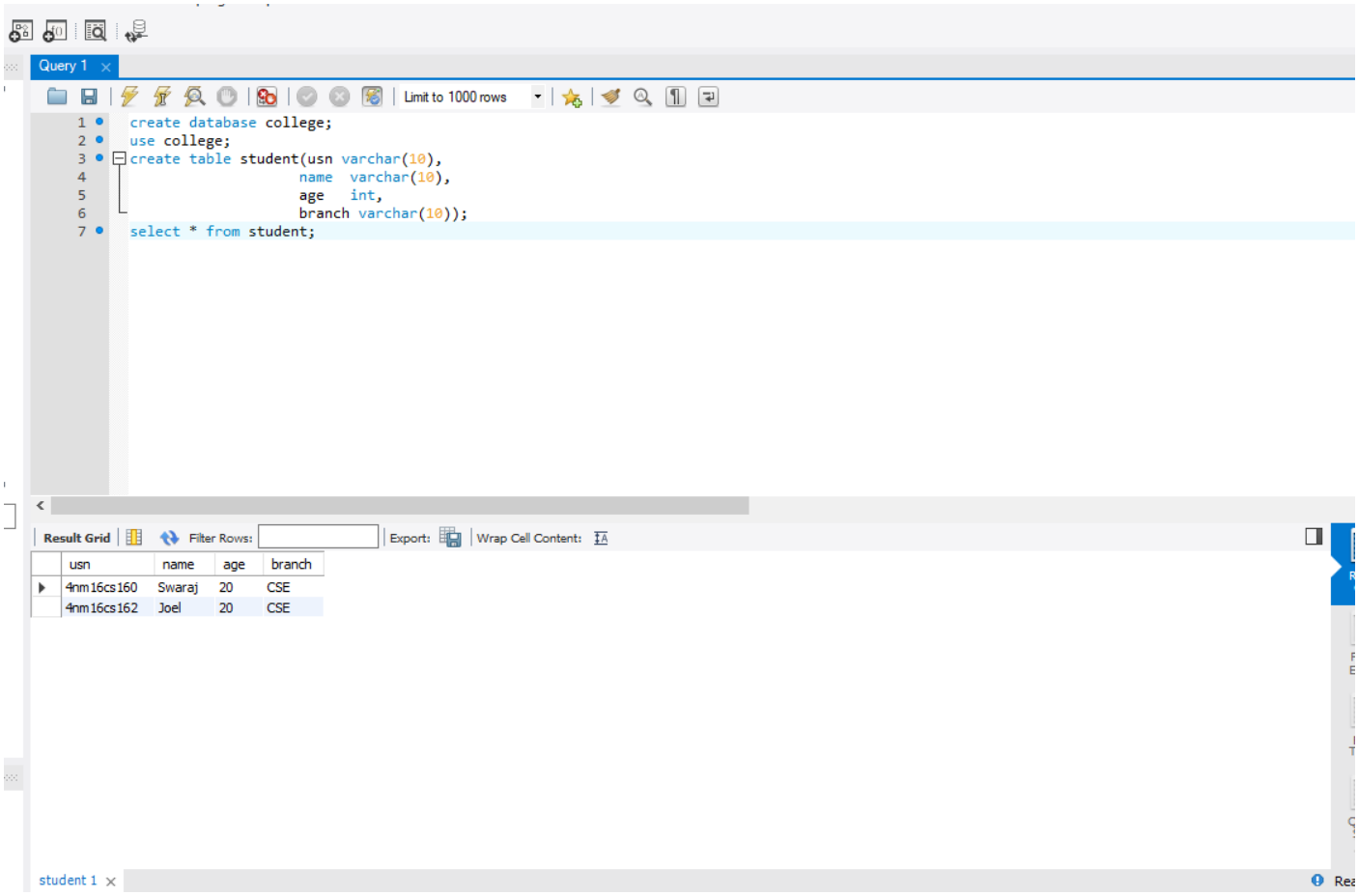
to create database

1.*goto command prompt>type* →**mysql –u root –p**

2.then type password---**root123** followed by **show databases;**

3.*create database and table as shown as below fig*

(I have used my sql workbench instead of command prompt)



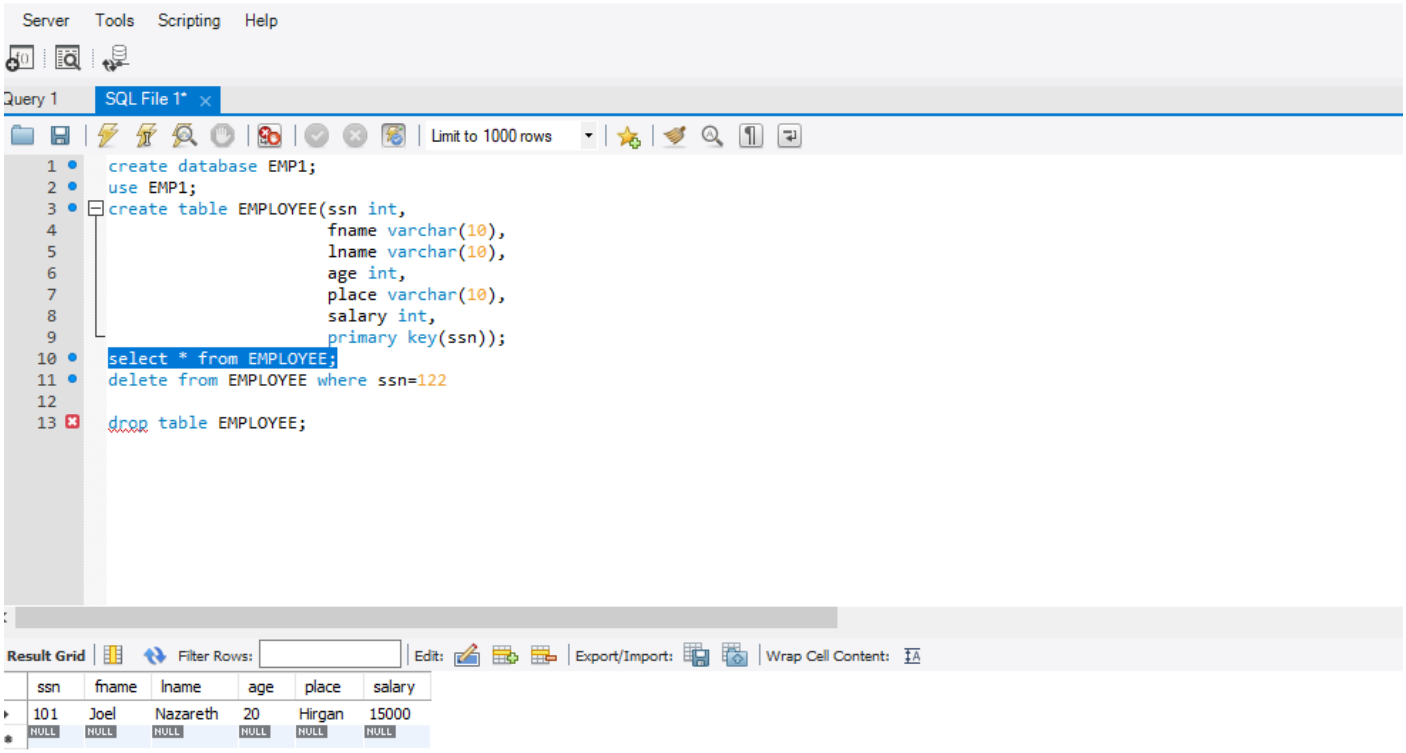
20. Develop a python program to perform the following operations on a database by considering the table Employee (SSN:Int, Fname: String, LName: String, Age:Int,Place:String,Salary:Int).Display the success and failure message.
a. Insert employee details
b. Delete a employee record
c. Update the employee details

```
'''
Created on 21-Oct-2018

@author: Terri
'''

import pymysql



def Insert():

    db = pymysql.connect("localhost","root","","EMP1" )
    cursor = db.cursor()
    ssn=int(input("Enter the ssn"));
    fname=input("Enter the first name");
    lname=input("Enter the last name");
```

```
    age=int(input("Enter the age"));
    place=input("Enter the place");
    salary=int(input("Enter the salary"));
    sql = "INSERT INTO EMPLOYEE VALUES
(%d,'%s','%s',%d,'%s',%d)"%(ssn,fname,lname,age,place,salary)
    try:
        cursor.execute(sql)
        db.commit()
        print("Successfully inserted")
    except:
        print("Unsuccessful")

    db.close()

def Delete():

    db = pymysql.connect("localhost","root","","EMP1" )
    cursor = db.cursor()
    ssn=int(input("Enter the SSN"))
    sql ="delete from EMPLOYEE where ssn=%d" %(ssn)
    try:
        cursor.execute(sql)
        db.commit()
        print("Deleted Successfully")
    except:
        print("error")
    db.close()
def Update():
    db = pymysql.connect("localhost","root","","EMP1" )
    cursor = db.cursor()
    ssn=int(input("Enter the ssn of the employee whose record to be updated "));
    fname=input("Enter the first name");
    lname=input("Enter the last name");
    age=int(input("Enter the age"));
    place=input("Enter the place");
    salary=int(input("Enter the salary"));
    try:
        cursor.execute("update EMPLOYEE set fname='%s' where ssn=%d"%(fname,ssn))
        db.commit()
        cursor.execute("update EMPLOYEE set lname='%s' where ssn=%d"%(lname,ssn))
        db.commit()
        cursor.execute("update EMPLOYEE set age=%d where ssn=%d"%(age,ssn))
        db.commit()
        cursor.execute("update EMPLOYEE set place='%s' where ssn=%d"%(place,ssn))
        db.commit()
        cursor.execute("update EMPLOYEE set salary=%d where ssn=%d"%(salary,ssn))
```

```
        db.commit()
        print("Successfully updated")
    except:
        print("Unsuccessful")
    db.close()

f=True
while(f):
    print("1.Insert\n2.Delete \n3.Update\n4.Exit")
    n=int(input("Enter the choice"))
    if(n==1):
        Insert()
    elif(n==2):
        Delete()
    elif(n==3):
        Update()
    elif(n==4):
        f=False
    else:
        print("Invalid choice")

    print()

    """
    output

    1.Insert
2.Delete
3.Update
4.Exit
Enter the choice1
Enter the ssn101
Enter the first nameJoel
Enter the last nameNazareth
Enter the age20
Enter the placeHirgan
Enter the salary15000
Successfully inserted

1.Insert
2.Delete
3.Update
4.Exit
Enter the choice3
Enter the ssn of the employee whose record to be updated 102
Enter the first nameSwaraj
```

```
Enter the last nameY>J
Enter the age23
Enter the placeManglur
Enter the salary12000
Successfully updated

1.Insert
2.Delete
3.Update
4.Exit
Enter the choice2
Enter the SSN102
Deleted Successfully

1.Insert
2.Delete
3.Update
4.Exit
Enter the choice4

"""
```



----------ALL THE BEST----------