**LAB-2**

**PYTHON**

**AUTHOR:**

Harsha Saranam-16230289

**OBJECTIVES:**

1. To find the books from the dictionary based on the input range given by the user.
2. To perform the operations with the contact list.
3. To design library management system using OOPS.
4. To find the maximum number of occurrences from the randomly generated set of numbers.

**FEATURES:**

1. We have the books with different prices, based on the range inputted by the user, the program results the list of books.
2. We have a list of operations to be performed:

* Display contact by name
* Display contact by number
* Edit contact by name
* Exit

1. This library management system performs actions regarding the students, faculties and books.

* Having 5 classes
* Performing inheritance
* Multiple inheritance
* Defined constructors

1. This program results the maximum number of occurrences of a number from the array generated randomly.

**CONFIGURATION:**

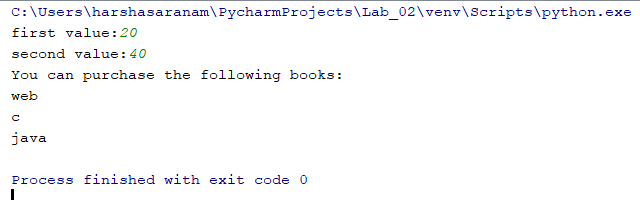
**Processor:** Intel Dual Core i7-6500U 2.5 GHz

**RAM:** 8 GB

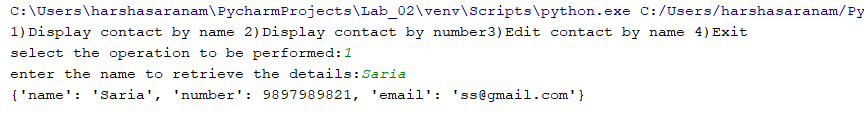
**SOFTWARE:** PYTHON IDE(PyCharm)

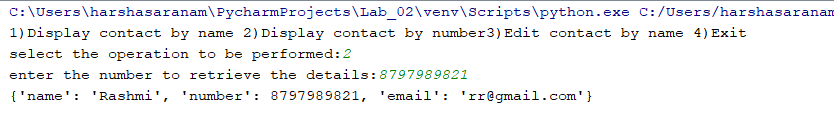
**INPUT/OUTPUT:**

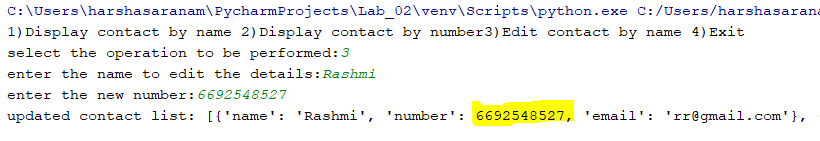
Program-1:

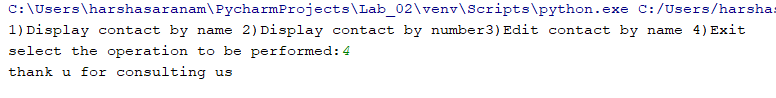


Program-2:

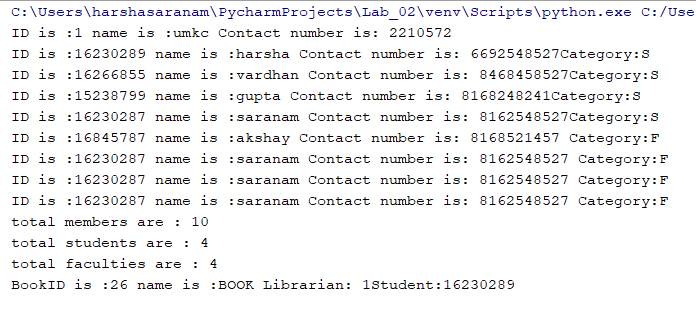




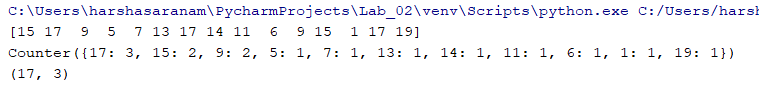




Program-3:



Program-4:



**IMPLEMENTATION:**

**Program-1:**

books={**'python'**:50,**'web'**:30,**'c'**:20,**'java'**:40}  
a=int(input(**"first value:"**)) *#inputting the first value*b=int(input(**"second value:"**)) *#inputting the second value*print(**"You can purchase the following books:"**)  
**for** key **in** books: *#taking only the keys fron the books* **if** books[key]>=a **and** books[key]<=b : *#based on the inputs given, it retieves the results* print(key)

**Program-2:**

**from** collections **import** defaultdict  
contact=[{**'name'**:**'Rashmi'**, **'number'**:8797989821, **'email'**:**'rr@gmail.com'**}, { **'name'**:**'Saria'**, **'number'**:9897989821, **'email'**: **'ss@gmail.com'**}]  
print(**"1)Display contact by name "  
 "2)Display contact by number"  
 "3)Edit contact by name "  
 "4)Exit "**) *#types of operation to be performed*a = int(input(**"select the operation to be performed:"**)) *#inputting the number for the operation to be performed***if** a==1:  
 name=str(input(**"enter the name to retrieve the details:"**))  
 **for** x **in** contact:  
 **if** x[**"name"**] == name: *#based on the name entered, it retrieves the user details* print(x)  
**elif** a==2:  
 num=int(input(**"enter the number to retrieve the details:"**))  
 **for** x **in** contact:  
 **if** x[**"number"**] == num: *#based on the number entered, it retrieves the user details* print(x)  
**elif** a==3:  
 name1=str(input(**"enter the name to edit the details:"**)) *#entering the name to which details to be edited* **for** x **in** contact:  
 **if** x[**"name"**] == name1:  
 num1=int(input(**"enter the new number:"**)) *#entering the new number to replace* x[**"number"**]=num1 *#replacing* print(**"updated contact list:"**,contact)  
  
**elif** a==4:  
 print(**"thank u for consulting us"**) *#exiting the system***else**:  
 print(**"enter a proper command"**)

**Program-3:**

**class** Member: *#class* count = 0  
 **def** \_\_init\_\_(self, name, id, mob): *#constructor* self.name = name  
 self.id = id  
 self.mob =mob  
 Member.count += 1  
  
 **def** get\_info(self):  
 **return "ID is :"**+ str(self.id)+ **" name is :"** + str(self.name)+**" Contact number is: "** +str(self.mob)  
  
  
**class** Librarian(Member): *#class #inheritance* count = 0  
 **def** \_\_init\_\_(self, name, id, mob): *#constructor* Member.\_\_init\_\_(self, name, id, mob) *#inherited data* **def** get\_info(self):  
 **return "ID is :"**+ str(self.id)+ **" name is :"** + str(self.name)+**" Contact number is: "** +str(self.mob)  
  
person=Librarian(**"umkc"**,**"1"**,**"2210572"**) *#instances for class librarian*person0=Librarian(**"umkc"**,**"2"**,**"1625378"**) *#instances for class librarian***class** Student(Member): *#class #inheritance* \_count=0  
 **def** \_\_init\_\_(self, name, id, mob): *#constructor* Member.\_\_init\_\_(self,name,id,mob) *#inherited data* self.type = **'S'** Student.\_count+=1  
 **def** get\_info(self):  
 **return "ID is :"**+ str(self.id)+ **" name is :"** + str(self.name)+**" Contact number is: "** +str(self.mob)+**"Category:"**+str(self.type)  
  
person1=Student(**"harsha"**,**"16230289"**,**"6692548527"**) *#instances for class student*person2=Student(**"vardhan"**,**"16266855"**,**"8468458527"**) *#instances for class student*person3=Student(**"gupta"**,**"15238799"**,**"8168248241"**) *#instances for class student*person4=Student(**"saranam"**,**"16230287"**,**"8162548527"**) *#instances for class student***class** Faculty(Member): *#class #inheritance* \_count=0  
 **def** \_\_init\_\_(self, name, id, mob): *#constructor* Member.\_\_init\_\_(self, name, id, mob) *#inherited data* self.type = **'F'** Faculty.\_count += 1  
 **def** get\_info(self):  
 **return "ID is :"** + str(self.id) + **" name is :"** + str(self.name) + **" Contact number is: "** + str(self.mob) + **" Category:"** + str(self.type)  
  
  
person5=Faculty(**"akshay"**,**"16845787"**,**"8168521457"**) *#instances for class faculty*person6=Faculty(**"saranam"**,**"16230287"**,**"8162548527"**) *#instances for class faculty*person7=Faculty(**"saranam"**,**"16230287"**,**"8162548527"**) *#instances for class faculty*person8=Faculty(**"saranam"**,**"16230287"**,**"8162548527"**) *#instances for class faculty*print(person.get\_info())  
print(person1.get\_info())  
print(person2.get\_info())  
print(person3.get\_info())  
print(person4.get\_info())  
print(person5.get\_info())  
print(person6.get\_info())  
print(person7.get\_info())  
print(person8.get\_info())  
  
print(**"total members are : "** + str(Member.count)) *#prints total number of ppl having their memberships*print(**"total students are : "** + str(Student.\_count)) *#total students*print(**"total faculties are : "** + str(Faculty.\_count)) *#total faculties***class** Books(Librarian,Student,Faculty):  
 **def** \_\_init\_\_(self, name, id, mob,Bookname,Bookid,studentid): *# constructor* Librarian.\_\_init\_\_(self, name, id, mob) *# inherited data* Student.\_\_init\_\_(self, name, id, mob) *# inherited data* self.Bookname=Bookname  
 self.Bookid = Bookid  
 self.studentid=studentid  
  
 **def** get\_info(self):  
 **return "BookID is :"** + str(self.Bookid) + **" name is :"** + str(self.Bookname) + **" Librarian: "** + str(  
 person.id) + **"Student:"** + str(person1.id)  
book1=Books(**"UMKC"**,1,2210572,**"BOOK"**,26,16230289)  
print(book1.get\_info())

**Program-4:**

**import** numpy **as** np *#importing***import** random  
**from** collections **import** Counter  
a=np.random.random\_integers(0,20,15) *#randomly taking 15 numbers from 0 to 20*print(a) *#printing the randomly generated numbers*print(Counter(a)) *#printing the each number and their frequency*print(Counter(a).most\_common(1)[0]) *#number with maximum number of occurances*