CS5590/490-Python/DeepLearning

ASSIGNMENT-2

Mohammad Mohsin Ali

16246618

Class ID: 26

**Author:**

**Objective:** Get familiar with python

**Features:**

The objective is to implement the lab task of finding book between range, implementing classes, finding most frequent number and create a contact list

**Configuration:**

Python 3.6.4

IDE: JetBrains PyCharm community Edition 2017.3.3

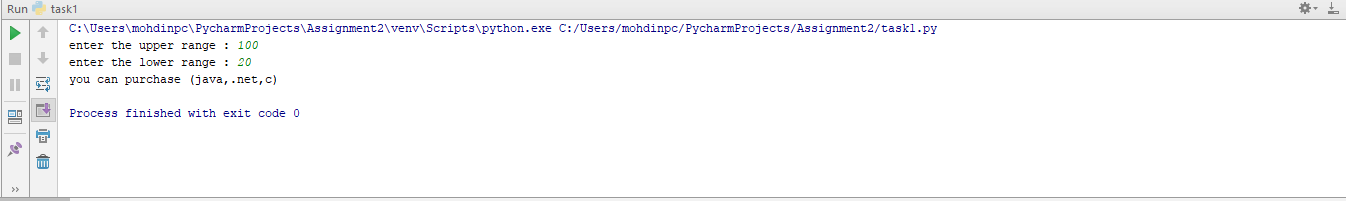
Numpy 1.14.0

**Input/output :**

**1**. find a book between price range.

**import** sys *#importing sys*max=sys.maxsize  
lib={**"python"**:10,**"java"**:20,**".net"**:30,**"c"**:40}*#dictionary taken*a=int(input(**"enter the upper range : "**))  
b=int(input(**"enter the lower range : "**))  
*#using simple if else statement to achieve output***if** a<0 **or** b<0:  
 print(**"we dont accept negitive price range"**)  
**elif** a<=10 **and** b>=0:  
 **if** a<10 **and** b>=0:  
 print(**"we do not have anything for you in this price range"**)  
 **else**:  
 print(**"you can purchase (%s)"**%list(lib.keys())[0])  
**elif** a<=20 **and** b>=0:  
 **if** a<=20 **and** b>=10:  
 **if** a<=20 **and** b>10:  
 **if** a<20 **and** b>10:  
 print(**"we do not have anything for you in this price range"**)  
 **else**:  
 print(**"you can purchase (%s)"**%list(lib.keys())[1])  
 **elif** a<20 **and** b>=10:  
 print(**"you can purchase (%s)"**%list(lib.keys())[0])  
 **else**:  
 print(**"you can purchase (%s,%s)"**%(list(lib.keys())[0],list(lib.keys())[1]))  
 **elif** a<20 **and** b>=0:  
 print(**"you can purchase (%s)"**%(list(lib.keys())[0]))  
 **else**:  
 print(**"you can purchase (%s,%s)"** % (list(lib.keys())[0], list(lib.keys())[1]))  
**elif** a<=30 **and** b>=0:  
 **if** a<=30 **and** b>=10:  
 **if** a<=30 **and** b>10:  
 **if** a<=30 **and** b>=20:  
 **if** a<=30 **and** b>20:  
 **if** a<30 **and** b>20:  
 print(**"we do not have anything for you in this price range"**)  
 **else**:  
 print(**"you can purchase (%s)"**%(list(lib.keys())[2]))  
 **elif** a<30 **and** b>=20:  
 print(**"you can purchase (%s)"**%(list(lib.keys())[1]))  
 **else**:  
 print(**"you can purchase (%s,%s)"**%(list(lib.keys())[1],list(lib.keys())[2]))  
 **else**:  
 print(**"you can purchase (%s,%s)"**%(list(lib.keys())[1],list(lib.keys())[2]))  
 **else**:  
 print(**"you can purchase (%s,%s,%s)"**%(list(lib.keys())[0],list(lib.keys())[1],list(lib.keys())[2]))  
 **elif** a<30 **and** b>=0:  
 print(**"you can purchase (%s,%s)"** % (list(lib.keys())[0], list(lib.keys())[1]))  
 **else**:  
 print(**"you can purchase (%s,%s,%s)"**%(list(lib.keys())[0],list(lib.keys())[1],list(lib.keys())[2]))  
**elif** a<=max **and** b>=0:  
 **if** a<=max **and** b>=10:  
 **if** a<=max **and** b>10:  
 **if** a<=max **and** b>=20:  
 **if** a<=max **and** b>20:  
 **if** a<=max **and** b>=30:  
 **if** a<=max **and** b>30:  
 **if** a<40 **and** b>30:  
 print(**"we do not have anything for you in this price range"**)  
 **else**:  
 print(**"you can purchase (%s)"**%(list(lib.keys())[3]))  
 **elif** a<40 **and** b>=30:  
 print(**"you can purchase (%s)"**%(list(lib.keys())[2]))  
 **else**:  
 print(**"you can purchase (%s,%s)"**%(list(lib.keys())[2],list(lib.keys())[3]))  
 **else**:  
 print(**"you can purchase (%s,%s)"**%(list(lib.keys())[2],list(lib.keys())[3]))  
 **else**:  
 print(**"you can purchase (%s,%s,%s)"**%(list(lib.keys())[1],list(lib.keys())[2],list(lib.keys())[3]))  
 **else**:  
 print(**"you can purchase (%s,%s,%s)"**%(list(lib.keys())[1],list(lib.keys())[2],list(lib.keys())[3]))  
 **else**:  
 print(**"you can purchase (%s,%s,%s,%s)"**%(list(lib.keys())[0],list(lib.keys())[1],list(lib.keys())[2],list(lib.keys())[3]))  
 **elif** a<40 **and** b>=0:  
 print(**"you can purchase (%s,%s,%s)"**%(list(lib.keys())[0],list(lib.keys())[1],list(lib.keys())[2]))  
 **else**:  
 print(**"you can purchase (%s,%s,%s,%s)"**%(list(lib.keys())[0],list(lib.keys())[1],list(lib.keys())[2],list(lib.keys())[3]))  
**else**:  
 print(**"we do not have anything for you in this price range"**)

we get output as,

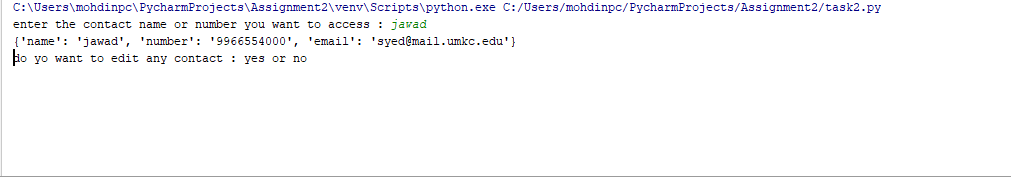


**2**.create a contact list and display contact by name ,number edit by name,exit

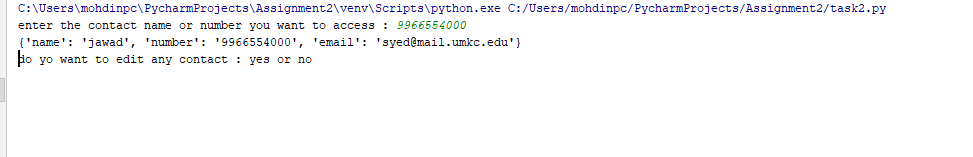
contact=[{**"name"**:**"jawad"**,**"number"**:**"9966554000"**,**"email"**:**"syed@mail.umkc.edu"**},{**"name"**:**"saad"**,**"number"**:**"9998866550"**,**"email"**:**"saad@mail.umkc.edu"**}]  
x=**""***#taking dictionaries inside the list***while** x!=**"exit"**:*#implementing while loop to meet reqirements,exit to exit* a=input(**"enter the contact name or number you want to access : "**)*#display contact by name or number* **if** a==contact[0][**"name"**] **or** a==contact[0][**"number"**]:  
 print(contact[0])  
 **elif** a==contact[1][**"name"**] **or** a==contact[1][**"number"**]:  
 print(contact[1])  
 **else**:  
 print(**"Sorry cannot match details entered "**)  
 b = input(**"do yo want to edit any contact : yes or no "**)*#edit contact* **if** b == **"yes"**:  
 c = input(**"enter the name of contact you want to edit : "**)  
 **if** c == contact[0][**"name"**]:  
 e = input(**"you can edit 'name','number','email'.type and enter anyone of thoose options : "**)  
 **if** e==**"name" or** e==**"number" or** e==**"email"**:  
 d = input(**"enter the new detail: "**)  
 contact[0][e] = d  
 print(**"the updated contact is "**,contact[0])  
 **else**:  
 print(**"enter the valid detail to edit"**)  
 **elif** c == contact[1][**"name"**]:  
 e = input(**"you can edit 'name','number','email'.type and enter anyone of thoose options : "**)  
 **if** e==**"name" or** e==**"number" or** e==**"email"**:  
 d = input(**"enter the new detail: "**)  
 contact[0][e] = d  
 print(**"the updated contact is "**, contact[1])  
 **else**:  
 print(**"enter the valid detail to edit"**)  
 **else**:  
 print(**"the name cannot be found"**)  
 **elif** b == **"no"**:  
 print(**"..........."**)  
 **else**:  
 print(**"enter the valid option or enter 'exit' to go back to main menu"**)  
 print(**"the updated contact list is"**,contact)  
 x=input(**"press 'exit' to close or any button to continue : "**)

the outputs are.

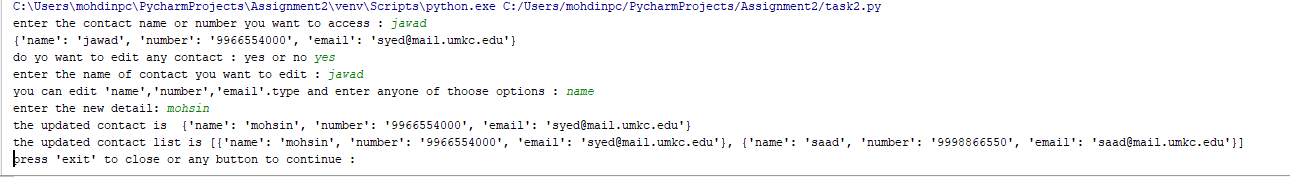
Display by number



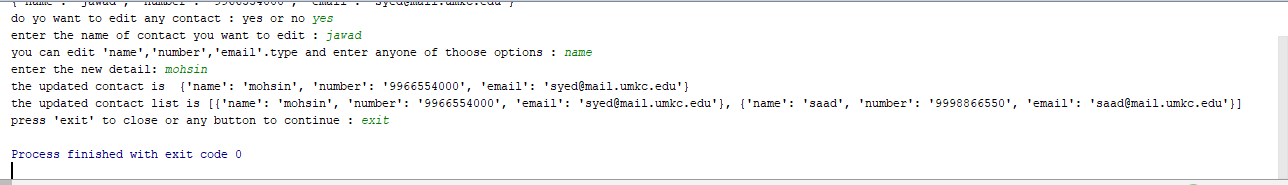
Display by phone



Edit by name



Exit

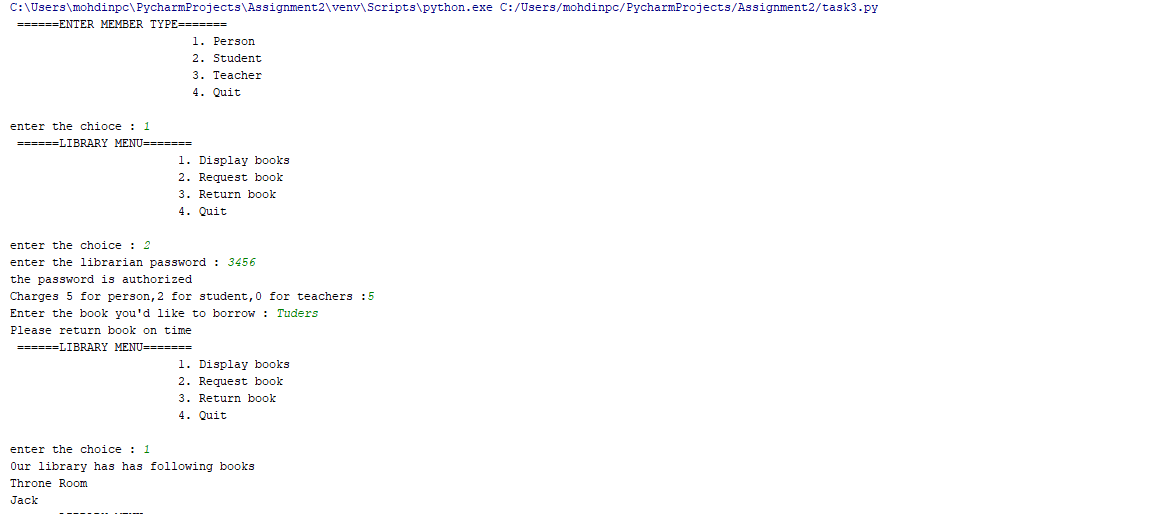


**3**.create a class

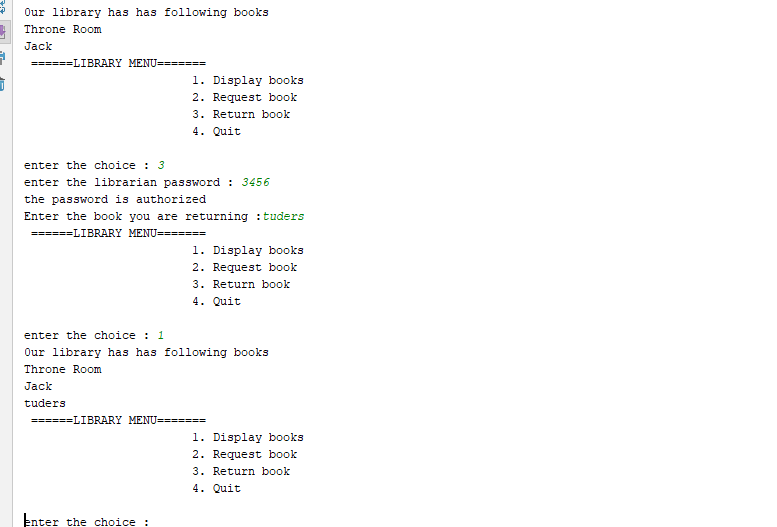
**import** sys  
**class** Library: *#first class* **def** \_\_init\_\_(self,list):*#as it is visible we have \_init\_ constructor in all class* self.list=list  
 **def** Availaibility(self):  
 print(**"0ur library has has following books"**)  
 **for** book **in** self.list:  
 print(book)  
 **def** lend(self,requested):  
 **if** requested **in** self.list:  
 print(**"Please return book on time"**)  
 self.list.remove(requested)  
 **else**:  
 print(**"The book is not available"**)  
 **def** returned(self,retrn):  
 self.list.append(retrn)  
  
**class** Person:*#2nd class* **def** \_\_init\_\_(self,a):*#\_init\_ constructor in second class* self.A=a  
 **def** RequestBook(self):  
 p = int(input(**"Charges 5 for person,2 for student,0 for teachers :"**))*#private data member* r=p+self.A  
 **if** r==5:  
 self.book=input(**"Enter the book you'd like to borrow : "**)  
 **return** self.book  
 **else**:  
 print(**"Pay the price that been asked according to type."**)  
 **def** ReturnBook(self):  
 self.book=input(**"Enter the book you are returning :"**)  
 **return** self.book  
**class** Student(Person):*#3rd class,inheriting from person class,super call constructor used* **def** \_\_init\_\_(self,b,x):*#\_init\_ constructor in third class* b+=x  
 Person.\_\_init\_\_(self,b)  
**class** Teacher(Person):*#4th class,inheriting from person class,mutiple inheritance* **def** \_\_init\_\_(self,c,y):*#\_init\_ constructor in fourth class* c+=y  
 Person.\_\_init\_\_(self,c)  
**class** Librarian():*#5th class* **def** \_\_init\_\_(self,d,e):*#\_init\_ constructor in fifth class* self.D=d  
 self.E=e  
 **def** Authorize(self):*#self is used in every class* self.D=int(input(**"enter the librarian password : "**))  
 **if** self.D==3456:  
 print(**"the password is authorized"**)  
 **else**:  
 print(**"sorry wrong password exiting..."**)  
 sys.exit()  
  
  
  
library=Library([**"Throne Room"**,**"Tuders"**,**"Jack"**])*#instance of class 1*a=0  
b=0  
c=0  
d=0  
e=0  
person=Person(a)*#instance of class 2*student=Student(b,3)*#instance of class 3*teacher=Teacher(c,5)*#instance of class 4*librarian=Librarian(d,e)*#instance of class 5*g=**False**h=**False  
while** h==**False**:  
 print(**""" ======ENTER MEMBER TYPE=======  
 1. Person  
 2. Student  
 3. Teacher  
 4. Quit  
 """**)  
 type=int(input(**"enter the chioce : "**))  
 **if** type==1:  
 **while** g==**False**:  
 print(**""" ======LIBRARY MENU=======  
 1. Display books  
 2. Request book  
 3. Return book  
 4. Quit  
 """**)  
 choice=int(input(**"enter the choice : "**))  
 **if** choice==1:  
 library.Availaibility()  
 **if** choice==2:  
 librarian.Authorize()  
 library.lend(person.RequestBook())*# implementing relationship between class 1 and 2* **if** choice==3:  
 librarian.Authorize()  
 library.returned(person.ReturnBook())*# implementing relationship between class 1 and 2* **if** choice==4:  
 print(**"going back to main menu...."**)  
 **break  
 if** type==2:  
 **while** g==**False**:  
 print(**""" ======LIBRARY MENU=======  
 1. Display books  
 2. Request book  
 3. Return book  
 4. Quit  
 """**)  
 choice=int(input(**"enter the choice : "**))  
 **if** choice==1:  
 library.Availaibility()  
 **if** choice==2:  
 librarian.Authorize()  
 library.lend(student.RequestBook())*# implementing relationship between class 1 and 3* **if** choice==3:  
 librarian.Authorize()  
 library.returned(student.ReturnBook())*# implementing relationship between class 1 and 3* **if** choice==4:  
 print(**"going back to main menu...."**)  
 **break  
 if** type==3:  
 **while** g==**False**:  
 print(**""" ======LIBRARY MENU=======  
 1. Display books  
 2. Request book  
 3. Return book  
 4. Quit  
 """**)  
 choice=int(input(**"enter the choice : "**))  
 **if** choice==1:  
 library.Availaibility()  
 **if** choice==2:  
 librarian.Authorize()  
 library.lend(teacher.RequestBook())*# implementing relationship between class 1 and 4* **if** choice==3:  
 librarian.Authorize()  
 library.returned(teacher.ReturnBook())*# implementing relationship between class 1 and 4* **if** choice==4:  
 print(**"going back to main menu...."**)  
 **break  
 if** type==4:  
 sys.exit()

the outputs are.

Request



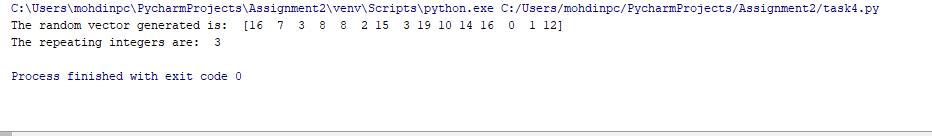
Return



**4**.most frequent integer using numpy.

**import** numpy **as** n*#importing numpy as n*x = n.random.randint(0,20, size= 15)*#initializing integers between 0 to 20 of size 15*print(**'The random vector generated is: '**,x)*#print the above logic*similar=n.bincount(x)*#counting the frequency of words*final= n.argmax(similar)*#most common integer*print(**'The repeating integers are: '**, final)*#printing*

the output is.



**Explanation:**

**1**. price range

There are may ways to implement this requirement, but the simplest way is loops

Initialize the dictionary first,’

lib={**"python"**:10,**"java"**:20,**".net"**:30,**"c"**:40}*#dictionary taken*a=int(input(**"enter the upper range : "**))  
b=int(input(**"enter the lower range : "**))

we take two inputs upper price and lower price.

Then we divide code into five parts,

**if** a<0 **or** b<0:

in this part we make sure that no negative number is being implement

**elif** a<=20 **and** b>=0:

all instances that we may face in this range

**elif** a<=30 **and** b>=0:

all instances that we may face in this range

**elif** a<=40 **and** b>=0:

all instances that we may face in this range

in the last case we make sure nothing except integers is being entered

**2**.contact list.

Here we are taking list inside a dictionary.

contact=[{**"name"**:**"jawad"**,**"number"**:**"9966554000"**,**"email"**:**"syed@mail.umkc.edu"**},{**"name"**:**"saad"**,**"number"**:**"9998866550"**,**"email"**:**"saad@mail.umkc.edu"**}]

then we take use of while loops,if else statements to print contact using name or number.

a=input(**"enter the contact name or number you want to access : "**)*#display contact by name or number***if** a==contact[0][**"name"**] **or** a==contact[0][**"number"**]:  
 print(contact[0])  
**elif** a==contact[1][**"name"**] **or** a==contact[1][**"number"**]:  
 print(contact[1])  
**else**:  
 print(**"Sorry cannot match details entered "**)

or edit

**if** b == **"yes"**:  
 c = input(**"enter the name of contact you want to edit : "**)  
 **if** c == contact[0][**"name"**]:  
 e = input(**"you can edit 'name','number','email'.type and enter anyone of thoose options : "**)  
 **if** e==**"name" or** e==**"number" or** e==**"email"**:  
 d = input(**"enter the new detail: "**)  
 contact[0][e] = d  
 print(**"the updated contact is "**,contact[0])  
 **else**:  
 print(**"enter the valid detail to edit"**)

and to exit we have given

**while** x!=**"exit"**:

**3**.creating a class.

**class** Library: *#first class* **def** \_\_init\_\_(self,list):*#as it is visible we have \_init\_ constructor in all class* self.list=list  
 **def** Availaibility(self):  
 print(**"0ur library has has following books"**)  
 **for** book **in** self.list:  
 print(book)  
 **def** lend(self,requested):  
 **if** requested **in** self.list:  
 print(**"Please return book on time"**)  
 self.list.remove(requested)  
 **else**:  
 print(**"The book is not available"**)  
 **def** returned(self,retrn):  
 self.list.append(retrn)  
  
**class** Person:*#2nd class* **def** \_\_init\_\_(self,a):*#\_init\_ constructor in second class* self.A=a  
 **def** RequestBook(self):  
 p = int(input(**"Charges 5 for person,2 for student,0 for teachers :"**))*#private data member* r=p+self.A  
 **if** r==5:  
 self.book=input(**"Enter the book you'd like to borrow : "**)  
 **return** self.book  
 **else**:  
 print(**"Pay the price that been asked according to type."**)  
 **def** ReturnBook(self):  
 self.book=input(**"Enter the book you are returning :"**)  
 **return** self.book  
**class** Student(Person):*#3rd class,inheriting from person class,super call constructor used* **def** \_\_init\_\_(self,b,x):*#\_init\_ constructor in third class* b+=x  
 Person.\_\_init\_\_(self,b)  
**class** Teacher(Person):*#4th class,inheriting from person class,mutiple inheritance* **def** \_\_init\_\_(self,c,y):*#\_init\_ constructor in fourth class* c+=y  
 Person.\_\_init\_\_(self,c)  
**class** Librarian():*#5th class* **def** \_\_init\_\_(self,d,e):*#\_init\_ constructor in fifth class* self.D=d  
 self.E=e  
 **def** Authorize(self):*#self is used in every class* self.D=int(input(**"enter the librarian password : "**))  
 **if** self.D==3456:

In this code we have taken five classes. The basic function of this code is to maintain the record of books taken. We have given Liberian class the authority to deny or approve the request class student and teacher are inherited from person. We have given logic that student pays less and teacher pays 0.we given list of book available and can be returned or requested from the librarian. We have given while looping **while** h==**False**:so that we can exit he program

**4**.using numpy to find the most frequent number.

**import** numpy **as** n

first we have import the numpy library and implement all the commands with it.

We use randit convert random values in to integerswe use bincount to count them and argmax to print the most repeated numbers.

All the outputs of above sections are given in input/output section.

**Deployment:**

1. Save the file in your computer
2. Install Python 3.6.4 and PyCharm IDE on your machine.
3. click open-files -location of the saved folder on pycharm.
4. Make sure the code running has .py extension.
5. Right click on the code screen and then click on run .
6. Enter input to see output.

**limitation**

1.Task 1 doesn’t take negative range.

2.input is not in integers in task 2

3.for certain logic in task 3 input should be in integers.

**References**

* <https://stackoverflow.com/>
* <https://www.tutorialspoint.com/python3/python_classes_objects.htm>