Python Basics

1.Write a program using operator overloading to input two complex and to find:

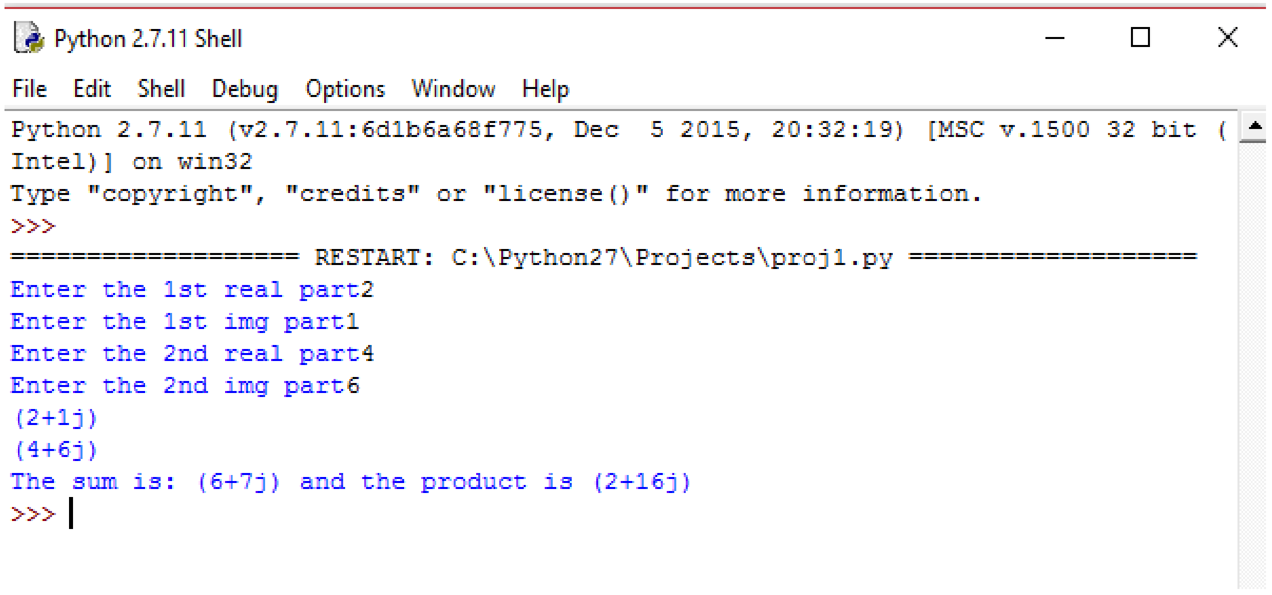
(a)Sum of two complex numbers

(b)Multiplication of the two numbers

Code:

```
a=int(raw_input("Enter the 1st real part"))
b=int(raw_input("Enter the 1st img part"))
c=int(raw_input("Enter the 2nd real part"))
d=int(raw_input("Enter the 2nd img part"))
e=complex(a,b)
f=complex(c,d)
print e
print f
g=e+f
h=e*f
print "The sum is:",g,"and the product is",h
```

Output:



```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=================== RESTART: C:\Python27\Projects\proj1.py ===================
Enter the 1st real part2
Enter the 1st img part1
Enter the 2nd real part4
Enter the 2nd img part6
(2+1j)
(4+6j)
The sum is: (6+7j) and the product is (2+16j)
>>>
```

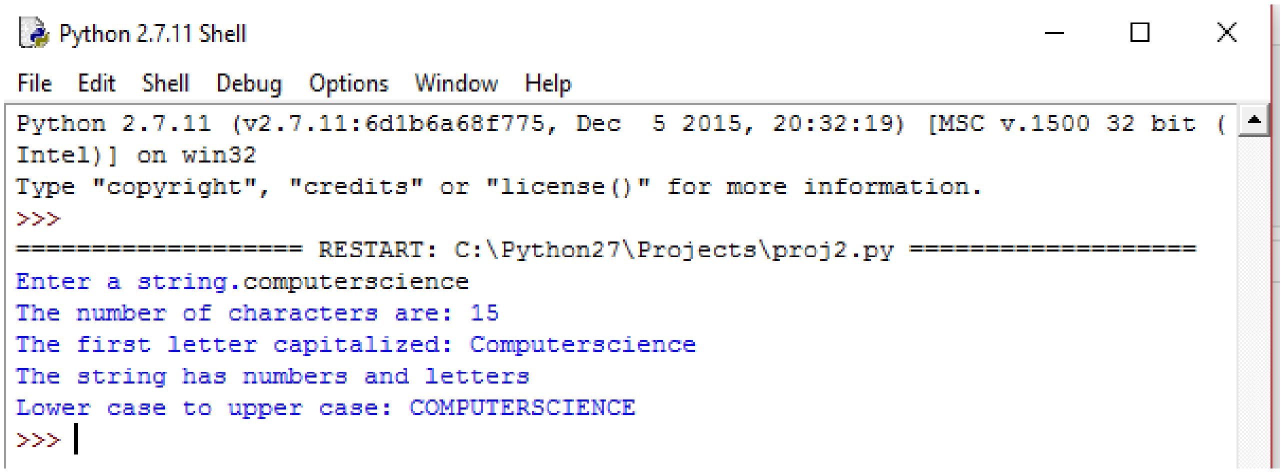
2.Write string method used to implement the following:

(a)To count the number of characters in the string.
(b)To change the first character in the string to capital letter.
(c)To check whether given character is letter or number.
(d)To change lower case to uppercase letter.

Code:

```
a=str(raw_input("Enter a string."))
b=len(a)
c=a.capitalize()
d=a.isalnum()
e=a.upper()
print"The number of characters are:",b
print"The first letter capitalized:",c
if d==True:
   print"The string has numbers and letters"
else:
   pass
print"Lower case to upper case:",e
```

Output:



```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
=================== RESTART: C:\Python27\Projects\proj2.py ===================
Enter a string.computerscience
The number of characters are: 15
The first letter capitalized: Computerscience
The string has numbers and letters
Lower case to upper case: COMPUTERSCIENCE
>>> 
```

3.Pay roll information system:

Declare the base class 'employee' with employee's number,name,designation,address,phone number.Define and declare the function getdata() and putdata() to get the employee's details and print employee's details.

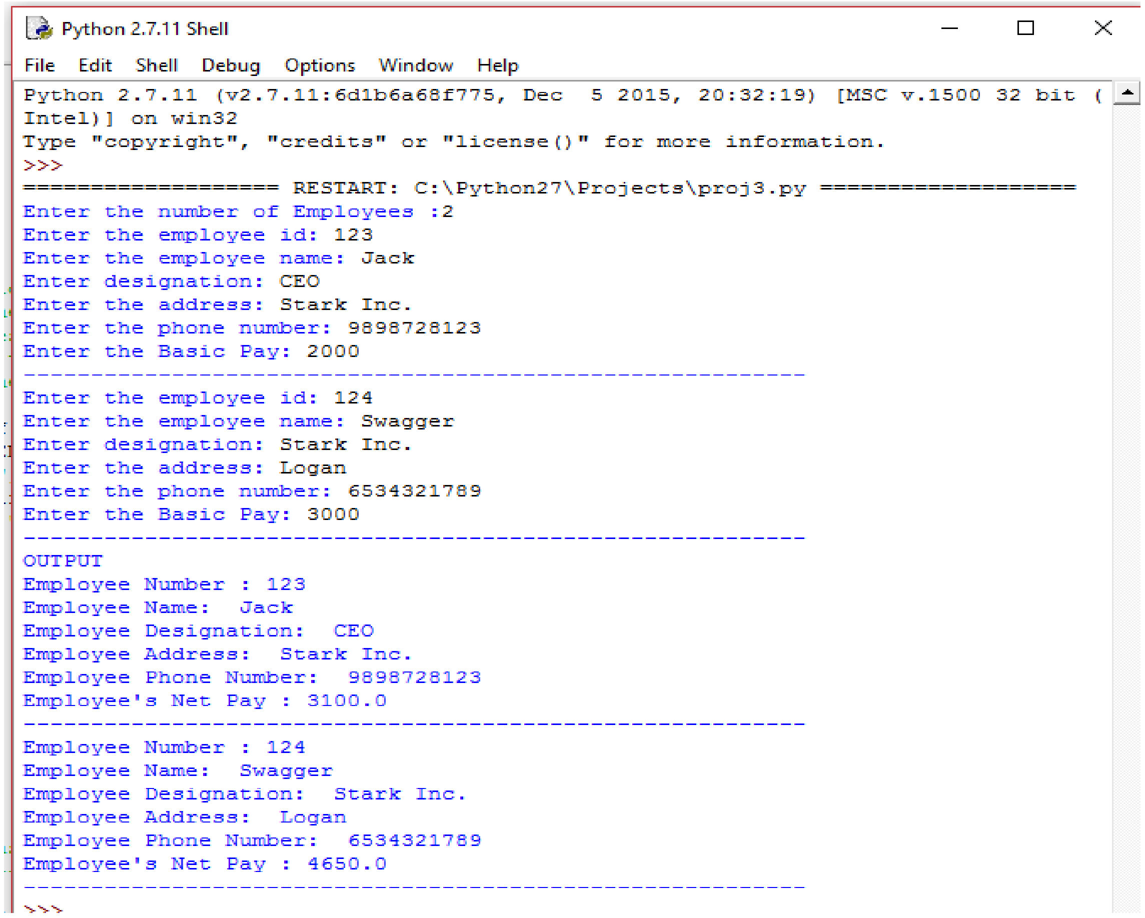
Declare the derived class salary with basic pay,DA,HRA,Gross pay,PF,Income tax and net pay. Declare define the function getdata1() to call getdata and get the basic pay. Define the function calculate() to find the net pay. Define the function display() to call putdata() and display salary details.

Create the derived class object. Read the number of employees. Call the function getdata1() and calculate() to each employee. Call the display() function.

Code:

```
class employee:
  def __init__(self):
    self.ENo=0
    self.EName=""
    self.Desig=""
    self.Address=""
    self.PNo=0
  def getdata(self):
    self.ENo=input("Enter the employee id: ")
    self.EName=raw_input("Enter the employee name: ")
    self.Desig=raw_input("Enter designation: ")
    self.Address=raw_input("Enter the address: ")
    self.PNo=input("Enter the phone number: ")
  def putdata(self):
    print "Employee Number :",self.ENo
    print "Employee Name: ",self.EName
    print "Employee Designation: ",self.Desig
    print "Employee Address: ",self.Address
    print "Employee Phone Number: ",self.PNo

class salary(employee):
  def __init__(self):
    employee.__init__(self)
    self.Basic=0
    self.DA=0
    self.HRA=0
    self.Gross=0
    self.PF=0
    selfIncomeTax=0
    self.NetPay=0
  def getdata1(self):
    employee.getdata(self)
    self.Basic=input("Enter the Basic Pay: ")
    print "-----------------------------------------------------------"
  def calculate(self):
        self.NetPay=self.Basic+(self.Basic*0.3)+(self.Basic*0.8)+(self.Basic*0.1)-(self.Basic*0.5)-
(self.Basic*0.15)
        return self.NetPay
    def display(self):
        employee.putdata(self)
        print "Employee's Net Pay :",self.calculate()
        print "-------------------------------------------------------"

n=int(raw_input("Enter the number of Employees :"))
global l1
l1=list()
for x in range (n):
    a=salary()
    l1.append(a)
    a.getdata1()
    a.calculate()
print "OUTPUT"
for i in l1:
    i.display()
```

Output:



```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
=================== RESTART: C:\Python27\Projects\proj3.py ===================
Enter the number of Employees :2
Enter the employee id: 123
Enter the employee name: Jack
Enter designation: CEO
Enter the address: Stark Inc.
Enter the phone number: 9898728123
Enter the Basic Pay: 2000
-----------------------------------------------------------
Enter the employee id: 124
Enter the employee name: Swagger
Enter designation: Stark Inc.
Enter the address: Logan
Enter the phone number: 6534321789
Enter the Basic Pay: 3000
-----------------------------------------------------------
OUTPUT
Employee Number : 123
Employee Name:  Jack
Employee Designation:  CEO
Employee Address:  Stark Inc.
Employee Phone Number:  9898728123
Employee's Net Pay : 3100.0
-----------------------------------------------------------
Employee Number : 124
Employee Name:  Swagger
Employee Designation:  Stark Inc.
Employee Address:  Logan
Employee Phone Number:  6534321789
Employee's Net Pay : 4650.0
-----------------------------------------------------------
>>> 
```

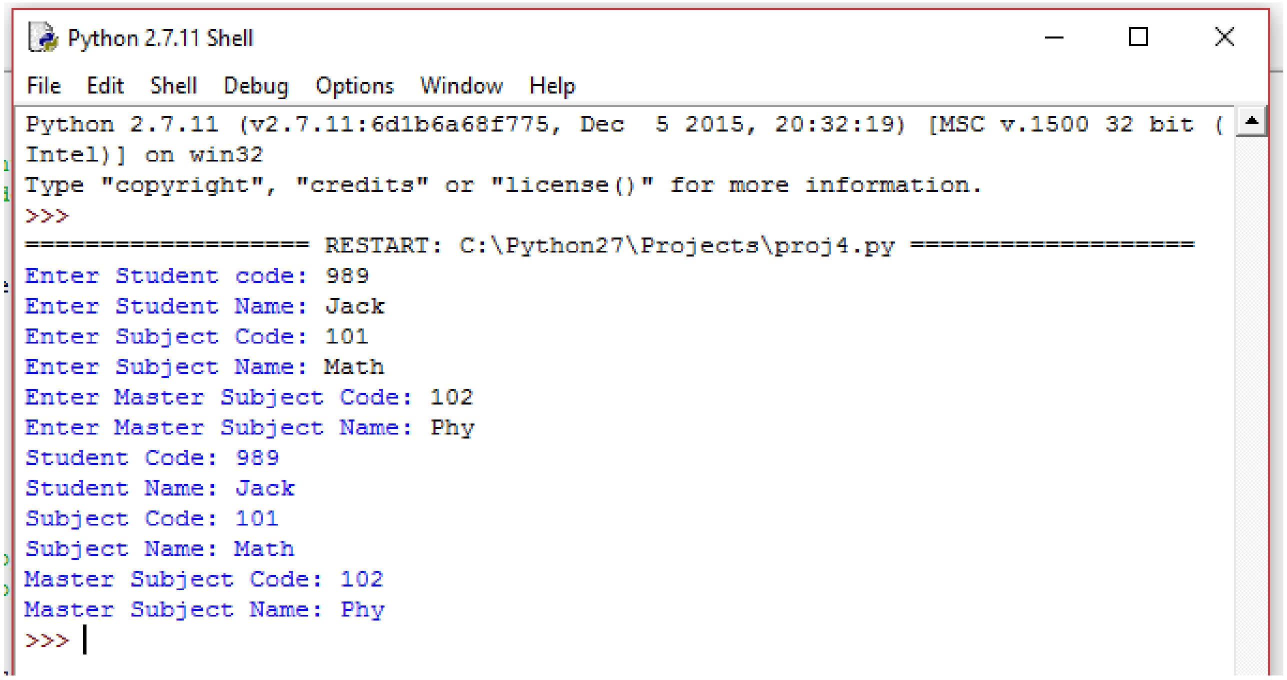
4.Implement the following using multilevel information. Create a student class with student number and name. Class graduate is created by using student. Graduate class is created using subject code and subject name. Class Post Graduate is created by using Graduate. Post Graduate class is created using master subject code and master subject name.

Code:

```
class Student:
    def __init__(self):
        self.StNo=0
        self.StName=""
    def readstud(self):
        self.StNo=raw_input("Enter Student code: ")
        self.StName=raw_input("Enter Student Name: ")
    def display(self):
        print "Student Code:",self.StNo
        print "Student Name:",self.StName

class Graduate(Student):
    def __init__(self):
        Student.__init__(self)
        self.Subcode=0
        self.Subname=""
    def readgrad(self):
        Student.readstud(self)
        self.Subcode=raw_input("Enter Subject Code: ")
        self.Subname=raw_input("Enter Subject Name: ")
    def display2(self):
        Student.display(self)
        print "Subject Code:",self.Subcode
        print "Subject Name:",self.Subname

class PostGraduate(Graduate,Student):
    def __init__(self):
        Graduate.__init__(self)
        Student.__init__(self)
        self.MasterSubcode=0
        self.MasterSubname=""
    def readpost(self):
        Graduate.readgrad(self)
        self.MasterSubcode=raw_input("Enter Master Subject Code: ")
        self.MasterSubname=raw_input("Enter Master Subject Name: ")
    def display3(self):
        Graduate.display2(self)
        print "Master Subject Code:",self.MasterSubcode
        print "Master Subject Name:",self.MasterSubname
```

a=PostGraduate()
a.readpost()
a.display3()

Output:



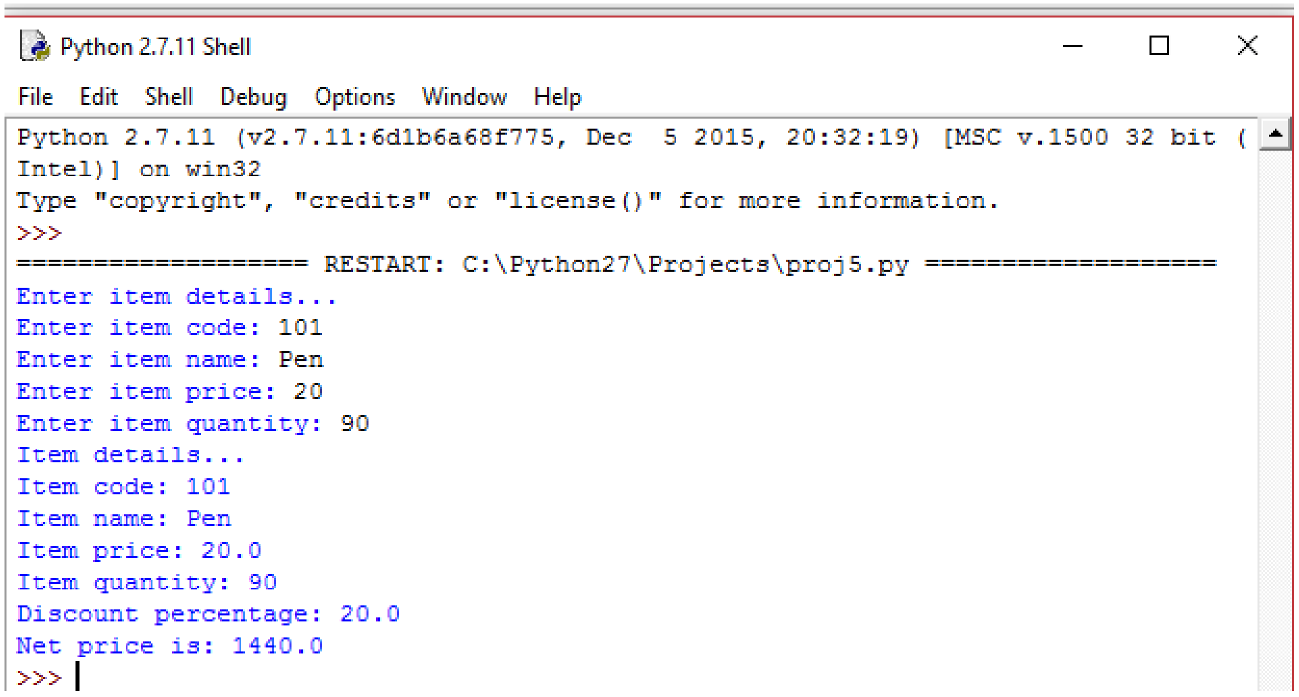
```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
=================== RESTART: C:\Python27\Projects\proj4.py ===================
Enter Student code: 989
Enter Student Name: Jack
Enter Subject Code: 101
Enter Subject Name: Math
Enter Master Subject Code: 102
Enter Master Subject Name: Phy
Student Code: 989
Student Name: Jack
Subject Code: 101
Subject Name: Math
Master Subject Code: 102
Master Subject Name: Phy
>>> 
```

5.Define a class ITEMINFO in Python with the following description:
Icode(Item Code), Item(Item Name), Price(Price of each item), Qty(Quantity in stock), Discount (Discount percentage on the item), Netprice(Final Price) Methods A member function FindDisc() to calculate discount as per the following rule: If Qty<=10 discount is zero. If Qty is from 11 to 20 discount is 15. If Qty>=20 discount is 20. A constructor(__init__method) to assign the value with 0 for Icode,Price,Qty,Netprice,discount and null for Item respectively. A function Buy() to allow user to enter values for Icode,Item,Price,Qty and call function FindDisc() to calculate the discount and Netprice(Price*(Quantity-Discount/100)). A function ShowAll() to allow user to view the content of all the data members.

Code:

```
class ITEMINFO:
    def __init__(self):
        self.ICode = 0
        self.IName = None
        self.Price = 0
        self.Qty = 0
        self.Discount = 0
        self.NetPrice = 0
    def Buy(self):
        print ("Enter item details...")
        self.ICode = int(raw_input("Enter item code: "))
        self.IName = raw_input("Enter item name: ")
        self.Price = float(raw_input("Enter item price: "))
        self.Qty = int(raw_input("Enter item quantity: "))
        self.Discount=self.FindDisc()
    def FindDisc(self):
        if self.Qty <= 10:
            self.Discount = 0.0
        elif self.Qty >= 11 and self.Qty <= 20:
            self.Discount = 15.0
        elif self.Qty > 20:
            self.Discount = 20.0
        self.NetPrice=((self.Price)*(self.Qty))-(((self.Price)*(self.Qty)*(self.Discount))/100)
        return self.Discount

    def ShowAll(self):
        print ("Item details...")
        print("Item code:", self.ICode)
        print("Item name:", self.IName)
        print("Item price:", self.Price)
        print("Item quantity:", self.Qty)
        print("Discount percentage:", self.Discount)
        print("Net price is:", self.NetPrice)
item=ITEMINFO()
item.Buy()
item.FindDisc()
item.ShowAll()
```

Output:

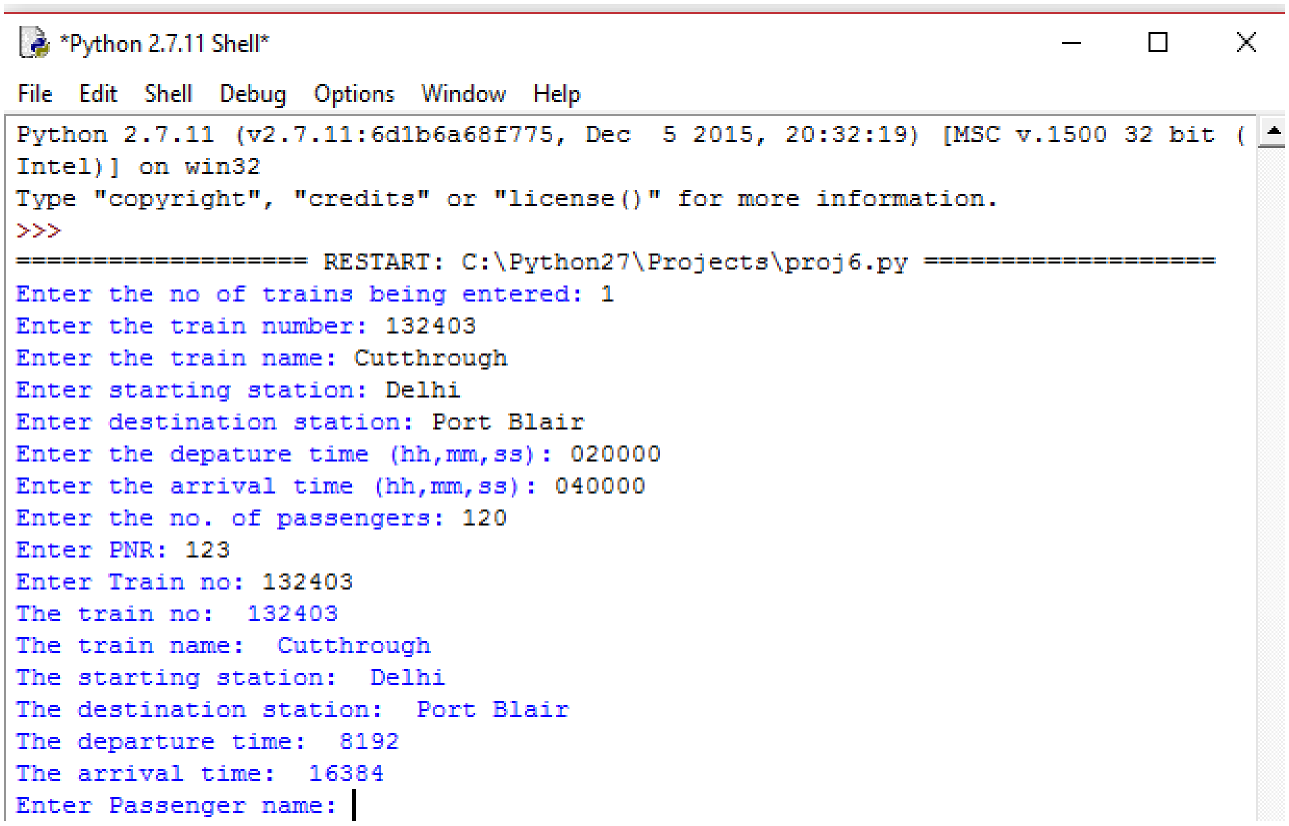


```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
=================== RESTART: C:\Python27\Projects\proj5.py ===================
Enter item details...
Enter item code: 101
Enter item name: Pen
Enter item price: 20
Enter item quantity: 90
Item details...
Item code: 101
Item name: Pen
Item price: 20.0
Item quantity: 90
Discount percentage: 20.0
Net price is: 1440.0
>>> 
```

6.Railway Reservation System: Declare the base class Train with train number,name,Starting Station, arrival time,etc. Define and declare the function getdata() and putdata() to get the train details and print the details. Deine search() function to search train detail using train number. Declare the derived class passenger with ticket number, PNR name of the passenger,gender, age, address,phone number, etc. Declare and define the function getdata1() to call search() function to get the train details and get the passenger's information. Define the function display() to call putdata() and display passenger's details. Create base class object. Read the number of trains. Create the derived class object. Read the number of passengers. Call the function getdata1() to each passenger. Call the display() function.

Code:
```
class Train:
   def __init__(self):
      self.TNo=0
      self.TName=""
      self.SStat=""
      self.DStat=""
      self.DTime=[]
      self.ATime=[]
      self.list=[]
      self.trainno=0
   def getdata(self):
      self.TNo=int(raw_input("Enter the train number: "))
      self.TName=raw_input("Enter the train name: ")
      self.SStat=raw_input("Enter starting station: ")
      self.DStat=raw_input("Enter destination station: ")
      self.DTime=eval(raw_input("Enter the depature time (hh,mm,ss): "))
      self.ATime=eval(raw_input("Enter the arrival time (hh,mm,ss): "))
   def putdata(self):
      print "The train no: ",self.TNo
      print "The train name: ",self.TName
      print "The starting station: ",self.SStat
      print "The destination station: ",self.DStat
      print "The departure time: ",self.DTime
      print "The arrival time: ",self.ATime
   @staticmethod
   def search(trainno):
      for z in l1:
         if z.TNo==trainno:
            z.putdata()
            break
      else:
         print "Train not found"

class Passenger(Train):
   def __init__(self):
      Train.__init__(self)
      self.PNR=0
      self.Tno=0
      self.Name=""
      self.gender=""
      self.age=0
      self.address=""
      self.Pno=0
      self.trainno
   def getdata1(self):
      self.PNR=input("Enter PNR: ")
      self.Tno=input("Enter Train no: ")
      Train.search(self.Tno)
      self.Name=raw_input("Enter Passenger name: ")
      self.gender=raw_input("Enter gender: ")
      self.age=input("Enter age:")
      self.address=raw_input("Enter address: ")
      self.Pno=input("Phone number: ")
   def display(self):
      for z in l1:
         if z.TNo==self.Tno:
            z.putdata()
      print "PNR of passenger: ",self.PNR
      print "Name of passenger: ",self.Name
      print "Gender: ",self.gender
      print "Passenger Age: ",self.age
      print "Passenger Address: ",self.address

l1=[]
l2=[]
tno=int(raw_input("Enter the no of trains being entered: "))
for x in range (tno):
   a=Train()
   l1.append(a)
   a.getdata()
pno=int(raw_input("Enter the no. of passengers: "))
for i in range (pno):
   b=Passenger()
   l2.append(b)
   b.getdata1()
for y in l2:
   y.display()
```

Output:



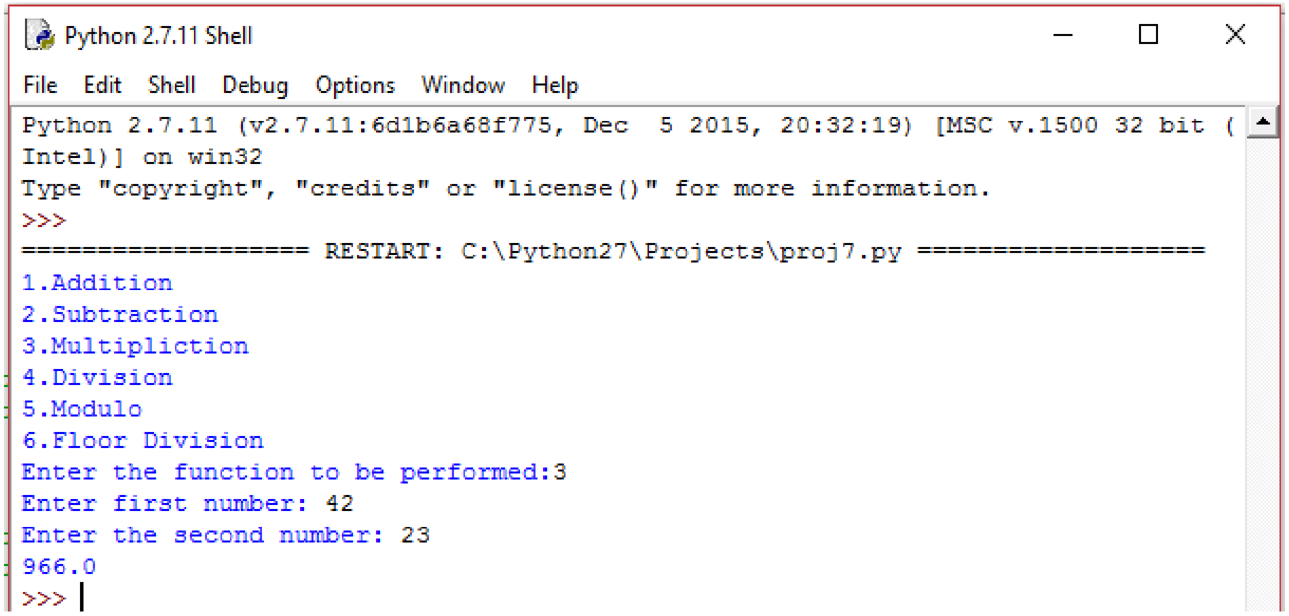
```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
=================== RESTART: C:\Python27\Projects\proj6.py ===================
Enter the no of trains being entered: 1
Enter the train number: 132403
Enter the train name: Cutthrough
Enter starting station: Delhi
Enter destination station: Port Blair
Enter the depature time (hh,mm,ss): 020000
Enter the arrival time (hh,mm,ss): 040000
Enter the no. of passengers: 120
Enter PNR: 123
Enter Train no: 132403
The train no:  132403
The train name:  Cutthrough
The starting station:  Delhi
The destination station:  Port Blair
The departure time:  8192
The arrival time:  16384
Enter Passenger name: 
```

7.Write a menu driven program to perform all mathematical operations using the concept of operator overloading.

Code:

```
class mathfunc:
   def __init__(self):
      self.sum=0.0
      self.dif=0.0
      self.pro=0.0
      self.quo=0.0
      self.rem=0.0
      self.floor=0.0
      self.num1=0.0
      self.num2=0.0
   def __add__(self):
      self.num1=float(raw_input("Enter first number: "))
      self.num2=float(raw_input("Enter the second number: "))
      self.sum=self.num1+self.num2
      print self.sum
   def __sub__(self):
      self.num1=float(raw_input("Enter the number to be subtracted: "))
      self.num2=float(raw_input("Enter the number from which you want to subtract: "))
      self.dif=self.num2-self.num1
      print self.dif
   def __mult__(self):
      self.num1=float(raw_input("Enter first number: "))
      self.num2=float(raw_input("Enter the second number: "))
      self.pro=self.num1*self.num2
      print self.pro
   def __div__(self):
      self.num1=float(raw_input("Enter the dividend : "))
      self.num2=float(raw_input("Enter the dividor: "))
      self.quo=self.num1/self.num2
      print self.quo
   def __mod__(self):
      self.num1=float(raw_input("Enter the dividend for modulo : "))
      self.num2=float(raw_input("Enter the dividor for modulo: "))
      self.rem=self.num1/self.num2
      print self.rem
   def __floor__(self):
      self.num1=float(raw_input("Enter first number: "))
      self.num2=float(raw_input("Enter the second number: "))
      decide=int(raw_input("Divide what by what? 1>num1/num2 <2>num2/num1: "))
      if decide==1:
         self.floor=self.num1/self.num2
      elif decide==2:
         self.floor=self.num2/self.num1
      print self.floor
   def __fundecide__(self):
      print "1.Addition"
      print "2.Subtraction"
      print "3.Multipliction"
      print "4.Division"
      print "5.Modulo"
      print "6.Floor Division"
      choice=int(raw_input("Enter the function to be performed:"))
      if choice==1:
         self.__add__()
      elif choice==2:
         self.__sub__()
      elif choice ==3:
         self.__mult__()
      elif choice==4:
         self.__div__()
      elif choice==5 :
         self.__mod__()
      else:
         self.__floor__()
a=mathfunc()
a.__fundecide__()
```

Output:



```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
=================== RESTART: C:\Python27\Projects\proj7.py ===================
1.Addition
2.Subtraction
3.Multipliction
4.Division
5.Modulo
6.Floor Division
Enter the function to be performed:3
Enter first number: 42
Enter the second number: 23
966.0
>>> 
```

8.Write a menu driven program to perform basic list operations.

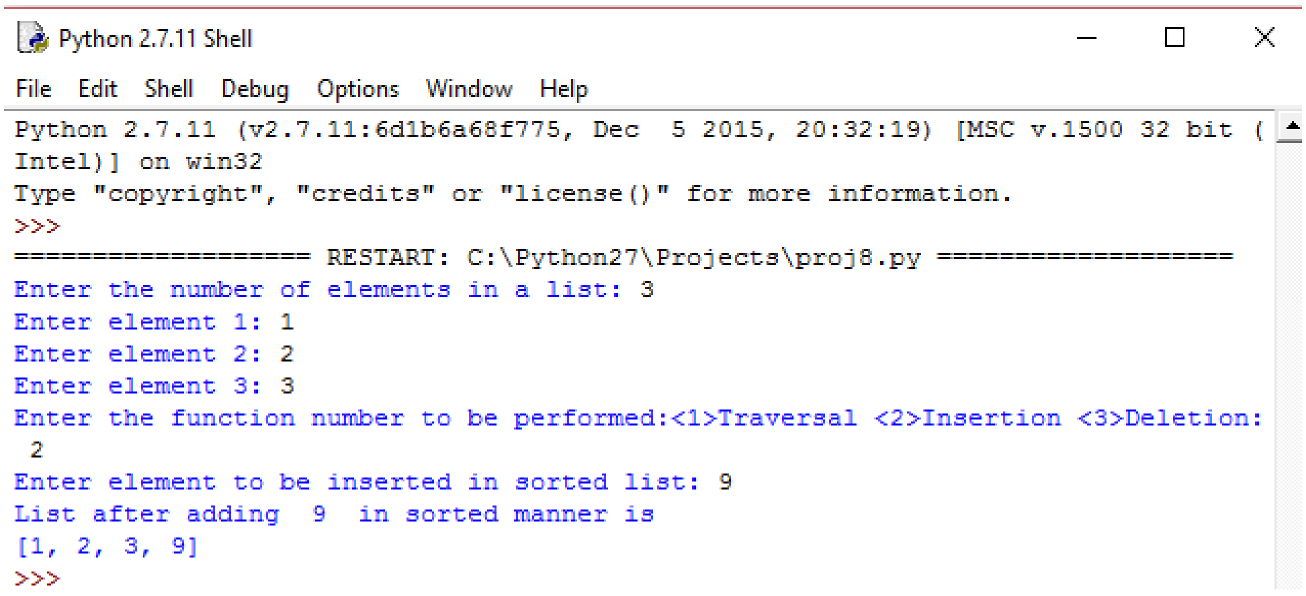
(a) Traversal in a list
(b) Insertion in a sorted list
(c) Deletion of an element from a list

Code:

```
import bisect

def traversal(List):
   size=len(List)
   for i in range(size):
      print List[i],
def insertion(List):
   Item=int(raw_input("Enter element to be inserted in sorted list: "))
   List.sort()
   bisect.insort(List,Item)
   print "List after adding ",Item," in sorted manner is "
   print List

def deletion(List,Item):
   beg=0
   last=len(List)-1
   while beg<=last:
      mid=(beg+last)/2
      if List[mid]==Item:
         return mid
      elif List[mid]>Item:
         beg=mid+1
      else:
         last=mid-1


N=int(raw_input("Enter the number of elements in a list: "))
List=[0]*N
for x in range(N):
   List[x]=int(raw_input("Enter element "+str(x+1)+": "))
choice=int(raw_input("Enter the function number to be performed:<1>Traversal <2>Insertion
<3>Deletion: "))
if choice==1:
   traversal(List)
elif choice==2:
   insertion(List)
else:
   Item=int(raw_input("Enter element to be deleted in list: "))
   index=int(deletion(List,Item))
   del List[index]
```

print "The list after deletion of element is: ",List

Output:



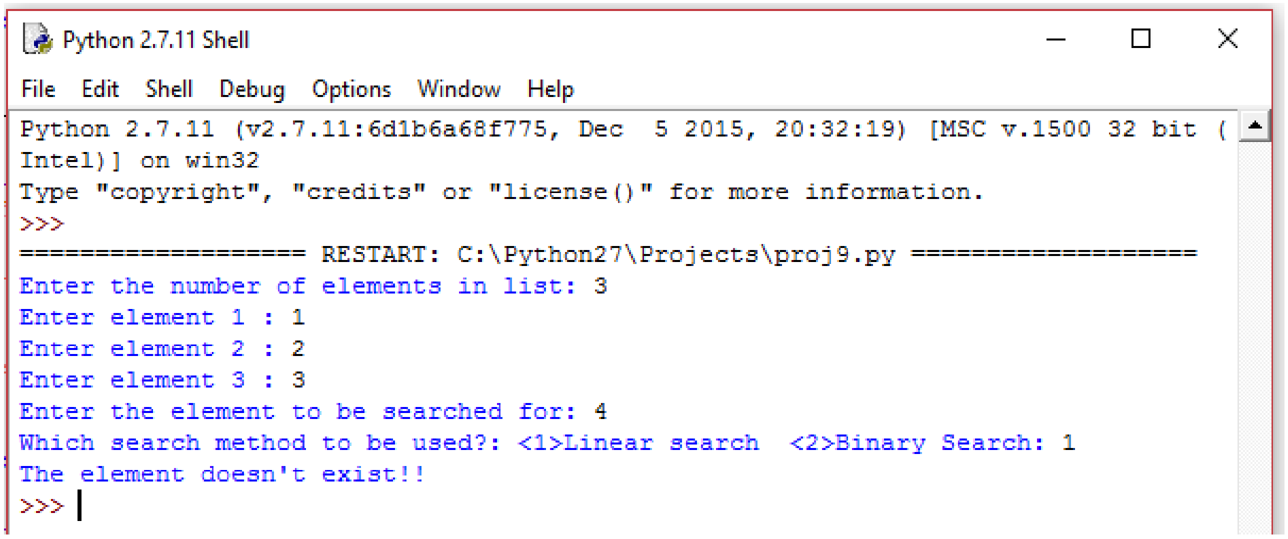
```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
=================== RESTART: C:\Python27\Projects\proj8.py ===================
Enter the number of elements in a list: 3
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Enter the function number to be performed:<1>Traversal <2>Insertion <3>Deletion:
 2
Enter element to be inserted in sorted list: 9
List after adding  9  in sorted manner is
[1, 2, 3, 9]
>>> 
```

9. Write a menu-driven program to perform searching techniques in a list string:
  (a) Linear search
  (b) Binary Search

Code:

```
def LSearch(List,Item):
   i=0
   while i<len(List) and List[i]<>Item:
      i+=1

   if i<len(List):
      if i==0:
         return "0"
      else:
         return i
   else:
      return False


def BSearch(List,Item):
   beg=0
   last=len(List)-1
   while beg<=last:
      mid=(beg+last)/2
      if  List[0]==Item:
         return "0"
      else:
         if List[mid]==Item:
            return mid
         elif List[mid]>Item:
            beg=mid+1
         else:
            last=mid-1

   else:
      return False

N=int(raw_input("Enter the number of elements in list: "))
List=[0]*N
for x in range(N):
   List[x]=int(raw_input("Enter element "+str(x+1)+" : "))
Item=int(raw_input("Enter the element to be searched for: "))
choice=int(raw_input("Which search method to be used?: <1>Linear search  <2>Binary Search: "))
if choice==1:
   index1=LSearch(List,Item)
   if index1:
        print "Found element at index ",int(index1), "or position ",int(index1)+1
    else:
        print "The element doesn't exist!!"
else:
    index2=BSearch(List,Item)
    if index2:
        print "Found element at index ",int(index2), "or position ",int(index2)+1
    else:
        print "Element doesn't exist"
```

Output:



```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
=================== RESTART: C:\Python27\Projects\proj9.py ===================
Enter the number of elements in list: 3
Enter element 1 : 1
Enter element 2 : 2
Enter element 3 : 3
Enter the element to be searched for: 4
Which search method to be used?: <1>Linear search  <2>Binary Search: 1
The element doesn't exist!!
>>> 
```

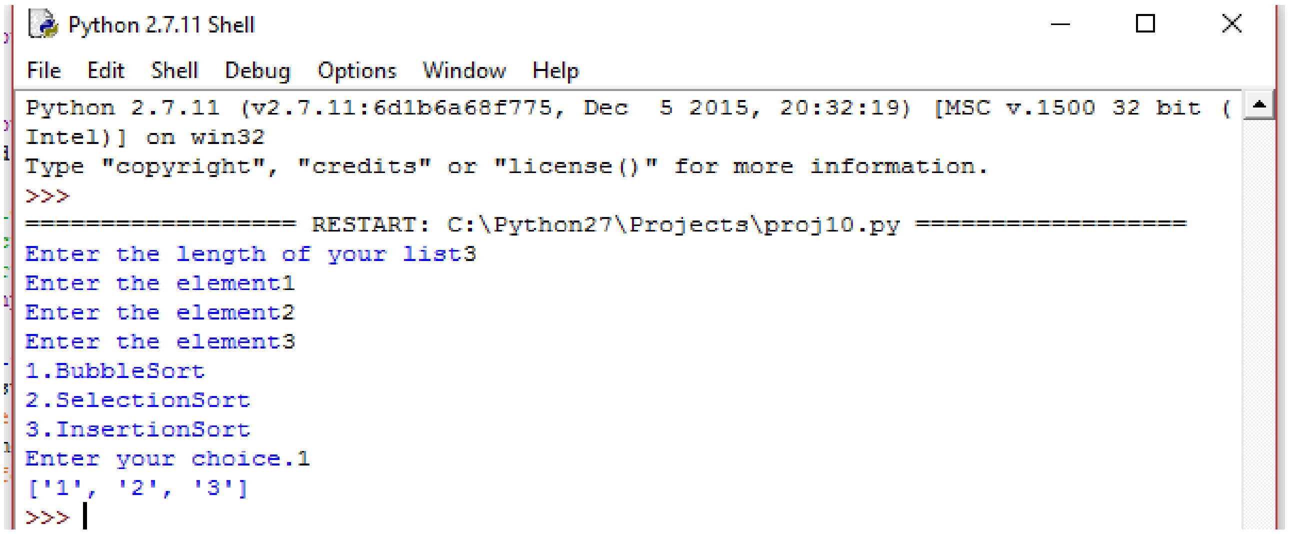
10. Write a menu-driven program to perform sorting of a list:

   (a) Selection Sort
   (b) Bubble Sort
   (c) Insertion Sort

<u>Code:</u>

```
L1=[]
n=int(raw_input("Enter the length of your list"))
for i<n:
   z=raw_input("Enter the element")
   L1.append(z)
   i=i+1
print"1.BubbleSort"
print"2.SelectionSort"
print"3.InsertionSort"
ch=int(raw_input("Enter your choice."))
if ch==1:
   def bubblesort(mylist):
      moreswaps=True
      while moreswaps:
         moreswaps=False
         for element in range(len(mylist)-1):
            if mylist[element]>mylist[element+1]:
               moreswaps=True
               temp=mylist[element]
               mylist[element]=mylist[element+1]
               mylist[element+1]=temp
      return mylist
   a=bubblesort(L1)
   print a
elif ch==2:
   def selectionsort(mylist):
      curpos=0
      for position in range(len(mylist)):
         minpos=position
         for scanpos in range(position+1,len(mylist)):
            if mylist[scanpos]<mylist[minpos]:
               minpos=scanpos
         temp=mylist[minpos]
         mylist[minpos]=mylist[curpos]
         mylist[curpos]=temp
      return mylist
   b=selectionsort(L1)
   print b
elif ch==3:
   def insertsort(mylist):
        for i in range(1,len(mylist)):
            v=L1[i]
            j=i
            while L1[j-i]>v and j>=1:
                L1[j]=L1[j-1]
                j=j-1
            L1[j]=v
        return mylist
    c=insertsort(L1)
    print c
else:
    print"Invalid Input"
```

Output:



```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
================== RESTART: C:\Python27\Projects\proj10.py ==================
Enter the length of your list3
Enter the element1
Enter the element2
Enter the element3
1.BubbleSort
2.SelectionSort
3.InsertionSort
Enter your choice.1
['1', '2', '3']
>>> 
```

11. Write a menu-driven program using list to perform the following operations

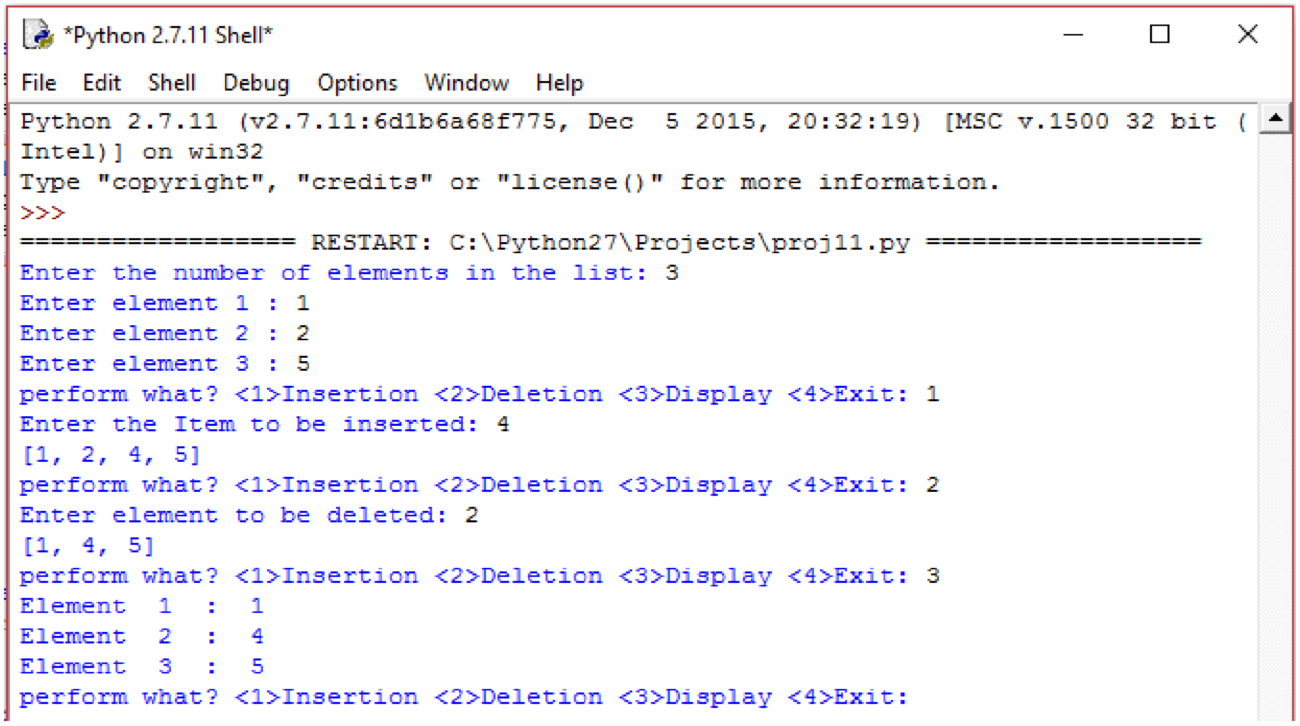
    (a)Insert an element
    (b)Delete an element
    (c)Display the contents of the list
    (d)Exit

Code:

```
import bisect
def insertion(List):
   Item=int(raw_input("Enter the Item to be inserted: "))
   bisect.insort(List,Item)
   print List
def deletion(List,Item):
   beg=0
   last=len(List)-1
   while beg<=last:
      if List[0]==Item:
         return "0"
      else:
         mid=(beg+last)/2
         if List[mid]==Item:
            return mid
         elif List[mid]>Item:
            beg=mid+1
         else:
            last=mid-1
def display(List):
   for x in range(len(List)):
      print "Element ",x+1," : ",List[x]

N=int(raw_input("Enter the number of elements in the list: "))
List=[0]*N
for x in range(N):
   List[x]=int(raw_input("Enter element "+str(x+1)+" : "))
while True:
   choice=int(raw_input("perform what? <1>Insertion <2>Deletion <3>Display <4>Exit: "))
   if choice==1:
      insertion(List)
   elif choice==2:
      Item=int(raw_input("Enter element to be deleted: "))
      index=deletion(List,Item)
      del List[index]
      print List
   elif choice==3:
      display(List)
   else:
```

    exit()

Output:



Python 2.7.11 Shell

File Edit Shell Debug Options Window Help

```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
================== RESTART: C:\Python27\Projects\proj11.py ==================
Enter the number of elements in the list: 3
Enter element 1 : 1
Enter element 2 : 2
Enter element 3 : 5
perform what? <1>Insertion <2>Deletion <3>Display <4>Exit: 1
Enter the Item to be inserted: 4
[1, 2, 4, 5]
perform what? <1>Insertion <2>Deletion <3>Display <4>Exit: 2
Enter element to be deleted: 2
[1, 4, 5]
perform what? <1>Insertion <2>Deletion <3>Display <4>Exit: 3
Element  1  :  1
Element  2  :  4
Element  3  :  5
perform what? <1>Insertion <2>Deletion <3>Display <4>Exit:
```

12.Write a program using lists in python to:

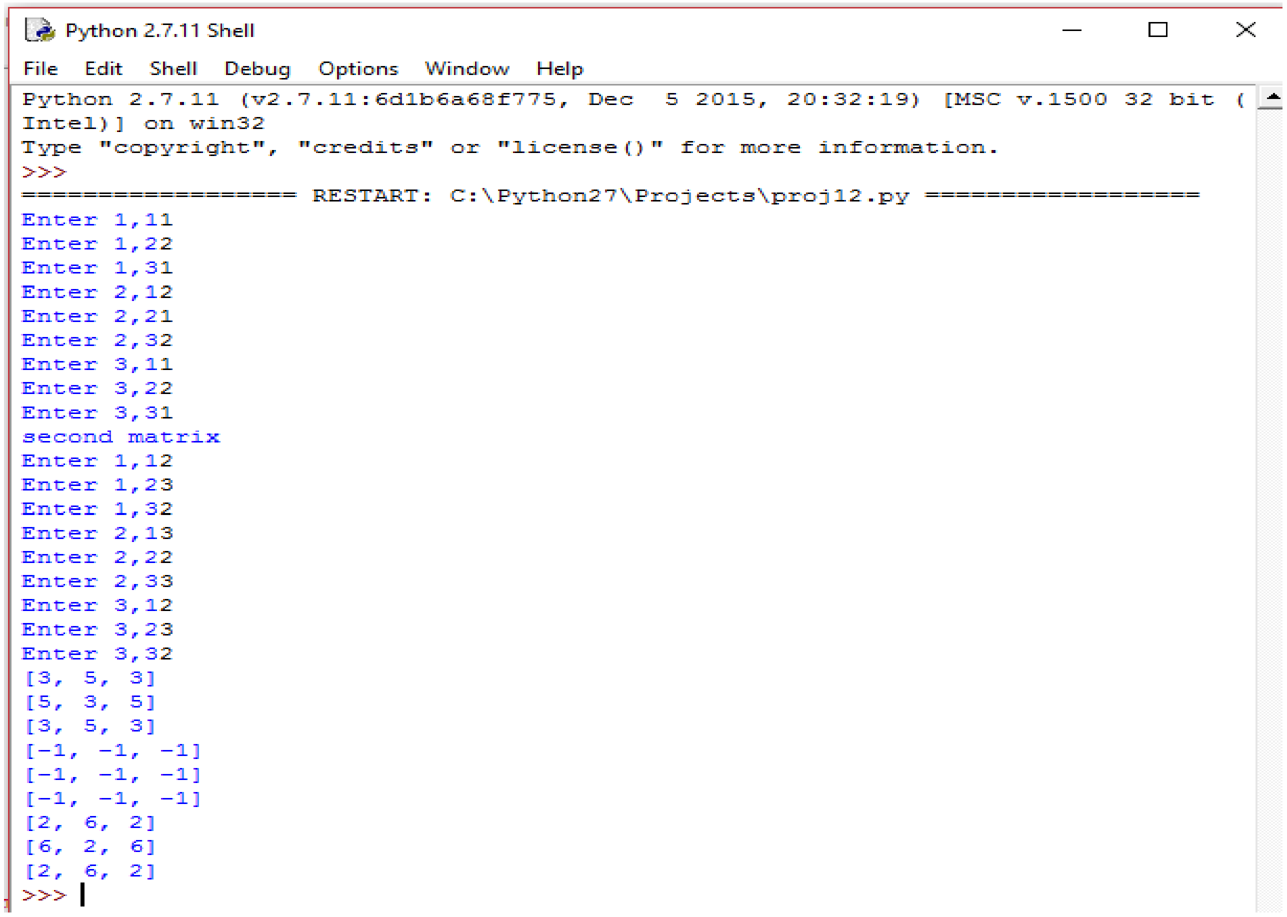
- a) Find the sum of two matrices
- b) Find the difference of two matrices
- c) Product of two matrices
- d) Exit

Code:

```
a=raw_input("Enter 1,1")
b=raw_input("Enter 1,2")
c=raw_input("Enter 1,3")
d=raw_input("Enter 2,1")
e=raw_input("Enter 2,1")
f=raw_input("Enter 2,1")
g=raw_input("Enter 3,1")
h=raw_input("Enter 3,1")
i=raw_input("Enter 3,1")
print "second matrix"
j=raw_input("Enter 1,1")
k=raw_input("Enter 1,1")
l=raw_input("Enter 1,1")
m=raw_input("Enter 2,1")
n=raw_input("Enter 2,1")
o=raw_input("Enter 2,1")
p=raw_input("Enter 3,1")
q=raw_input("Enter 3,1")
r=raw_input("Enter 3,1")

X = [[a,b,c],
   [d,e,f],
   [g,h,i]]

Y = [[j,k,l],
   [m,n,o],
   [p,q,r]]

result = [[0,0,0],
        [0,0,0],
        [0,0,0]]

# iterate through rows
for i in range(len(X)):
   # iterate through columns
   for j in range(len(X[0])):
       result[i][j] = int(X[i][j]) + int(Y[i][j])
```

```
for r in result:
  print(r)

# iterate through rows
for i in range(len(X)):
  # iterate through columns
  for j in range(len(X[0])):
     result[i][j] = int(X[i][j]) - int(Y[i][j])

for r in result:
  print(r)

# iterate through rows
for i in range(len(X)):
  # iterate through columns
  for j in range(len(X[0])):
     result[i][j] = int(X[i][j]) * int(Y[j][i])

for r in result:
  print(r)
```

Output:



```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
================== RESTART: C:\Python27\Projects\proj12.py ==================
Enter 1,11
Enter 1,22
Enter 1,31
Enter 2,12
Enter 2,21
Enter 2,32
Enter 3,11
Enter 3,22
Enter 3,31
second matrix
Enter 1,12
Enter 1,23
Enter 1,32
Enter 2,13
Enter 2,22
Enter 2,33
Enter 3,12
Enter 3,23
Enter 3,32
[3, 5, 3]
[5, 3, 5]
[3, 5, 3]
[-1, -1, -1]
[-1, -1, -1]
[-1, -1, -1]
[2, 6, 2]
[6, 2, 6]
[2, 6, 2]
>>> 
```

13.Write a program to implement Stack Operations as a list

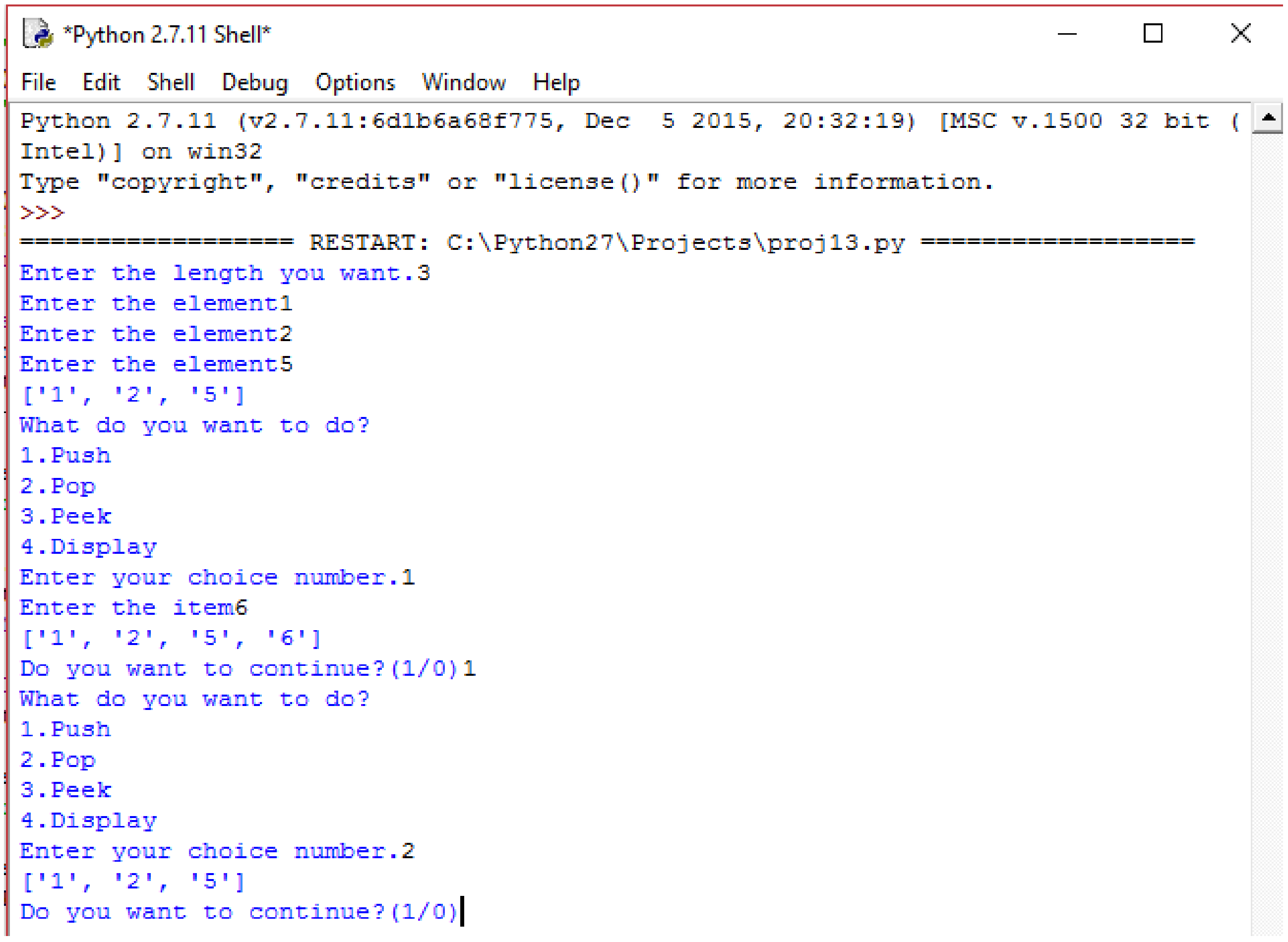
a) push
b) pop
c)peek
d) Display Stack
e)Exit

Code:

```
L1=[]
n=int(raw_input("Enter the length you want."))
for i in range(n):
   a=raw_input("Enter the element")
   L1.append(a)
print L1
def is_empty(stk):
   if stk == []:
      return True
   else:
      return False
def Push(stk,item):
   stk.append(item)
   top=len(stk)+ 1
def Pop(stk):
   if is_empty(stk):
      print"Underflow"
   else:
      item=stk.pop()
      if len(stk)==0:
         top=None
      else :
         top=len(stk)-1
      return item
def Peek(stk):
   if is_empty(stk):
      print"Underflow"
   else:
      top=len(stk)-1
      print stk[top]

def Display(stk):
   for i in range(n):
      print L1[i]
ans=1
while ans==1:
   print"What do you want to do?"
   print"1.Push"
```

```
    print"2.Pop"
    print"3.Peek"
    print"4.Display"
    ch=int(raw_input("Enter your choice number."))
    if ch==1:
        x=raw_input("Enter the item")
        Push(L1,x)
        print L1
    elif ch==2:
        Pop(L1)
        print L1
    elif ch==3:
        Peek(L1)
        print L1
    elif ch==4:
        Display(L1)
    else:
        print"Invalid input"
    ans=int(raw_input("Do you want to continue?(1/0)"))
```

Output:



```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
================== RESTART: C:\Python27\Projects\proj13.py ==================
Enter the length you want.3
Enter the element1
Enter the element2
Enter the element5
['1', '2', '5']
What do you want to do?
1.Push
2.Pop
3.Peek
4.Display
Enter your choice number.1
Enter the item6
['1', '2', '5', '6']
Do you want to continue?(1/0)1
What do you want to do?
1.Push
2.Pop
3.Peek
4.Display
Enter your choice number.2
['1', '2', '5']
Do you want to continue?(1/0)
```

14.Write a program to implement Queue Operations as a list

a) Enqueue

b) Dequeue

c)peek

d) Display Queue

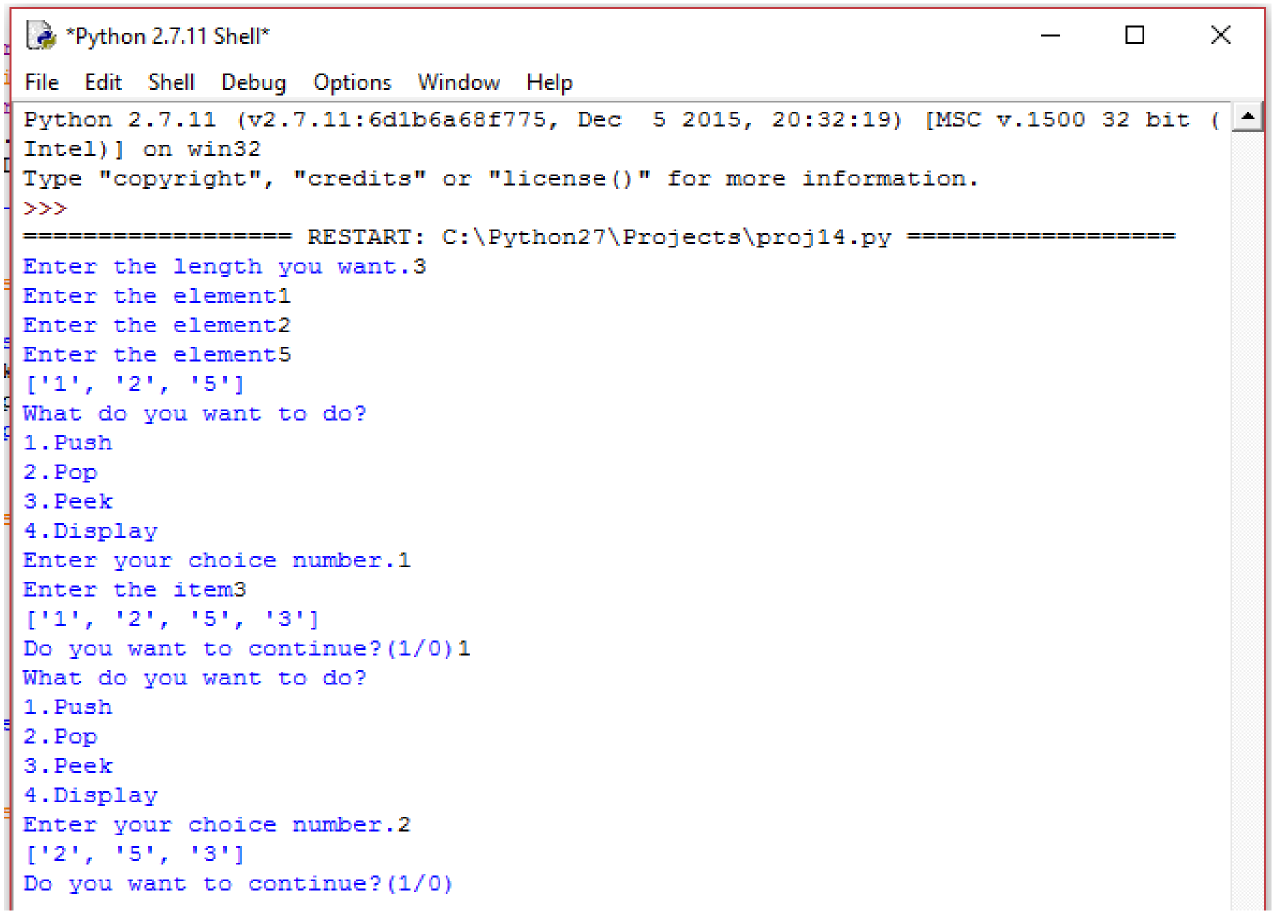
e)Exit

Code:

```
L1=[]
n=int(raw_input("Enter the length you want."))
for i in range(n):
   a=raw_input("Enter the element")
   L1.append(a)
print L1
def is_empty(stk):
   if stk == []:
      return True
   else:
      return False
def Push(stk,item):
   stk.append(item)
```

```
    top=len(stk)+ 1
def Pop(stk):
    if is_empty(stk):
        print"Underflow"
    else:
        item=stk.pop(0)
        if len(stk)==0:
            top=None
        else :
            top=len(stk)-1
        return item
def Peek(stk):
    if is_empty(stk):
        print"Underflow"
    else:
        top=len(stk)-1
        print stk[top]

def Display(stk):
    for i in range(n):
        print L1[i]
ans=1
while ans==1:
    print"What do you want to do?"
    print"1.Push"
    print"2.Pop"
    print"3.Peek"
    print"4.Display"
    ch=int(raw_input("Enter your choice number."))
    if ch==1:
        x=raw_input("Enter the item")
        Push(L1,x)
        print L1
    elif ch==2:
        Pop(L1)
        print L1
    elif ch==3:
        Peek(L1)
        print L1
    elif ch==4:
        Display(L1)
    else:
        print"Invalid input"
    ans=int(raw_input("Do you want to continue?(1/0)"))
```

Output:


Python 2.7.11 Shell

```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: C:\Python27\Projects\proj14.py ==================
Enter the length you want.3
Enter the element1
Enter the element2
Enter the element5
['1', '2', '5']
What do you want to do?
1.Push
2.Pop
3.Peek
4.Display
Enter your choice number.1
Enter the item3
['1', '2', '5', '3']
Do you want to continue?(1/0)1
What do you want to do?
1.Push
2.Pop
3.Peek
4.Display
Enter your choice number.2
['2', '5', '3']
Do you want to continue?(1/0)
```

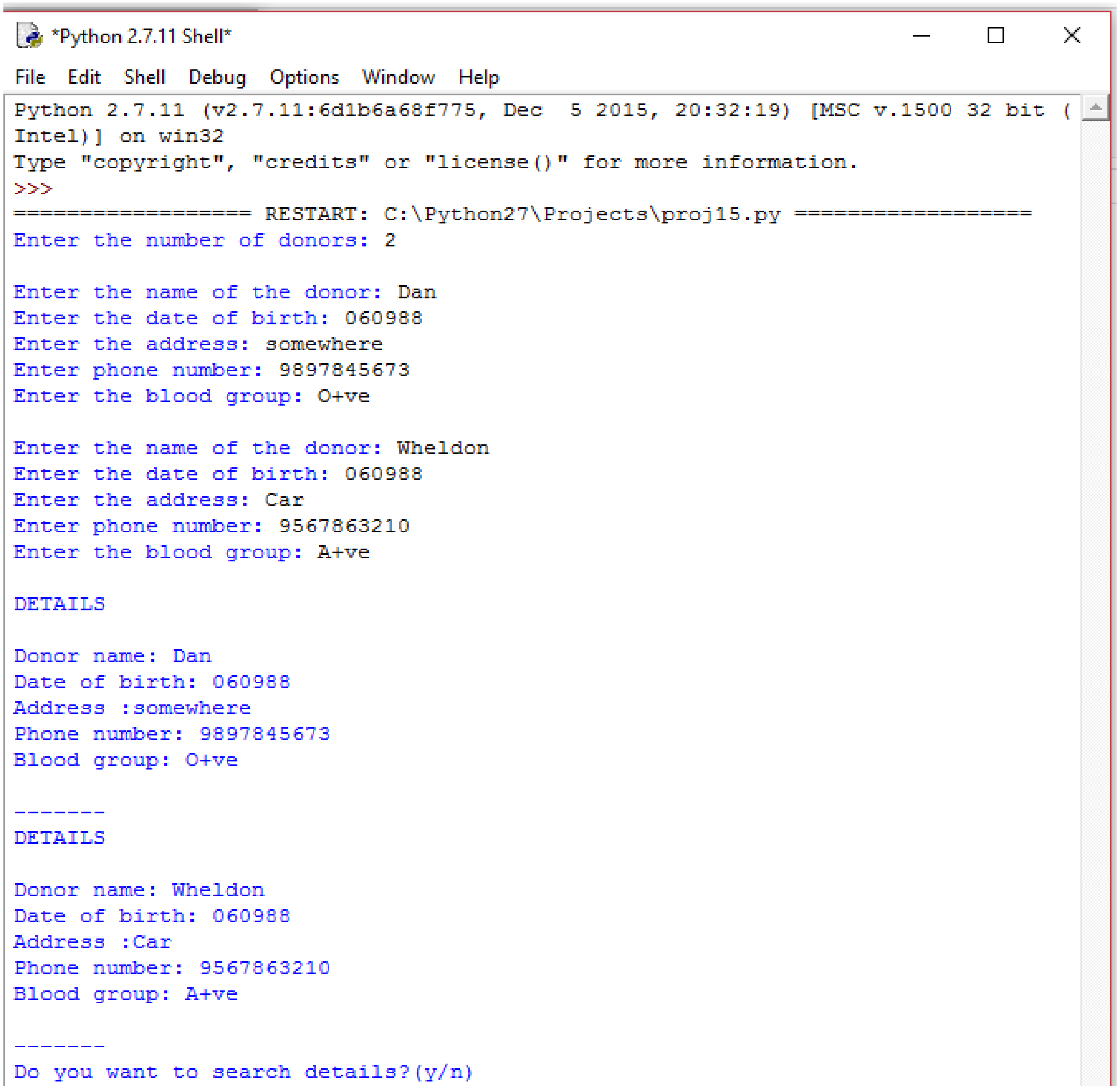
15. Define the class with following specification.
Class Donor
Private members:
Name, Date of birth, address, phone number, blood group
Public members:
Input()(create the file of donor)
Append()(append a record in the file)
Show()(display the record)
Search()(given the blood group displays name, address of the donor)
All the data must be stored in a file

Code:

```
class Donor:
   def __init__(self):
      self.__name=''
      self.__DOB=''
      self.__address=''
      self.__pno=0
      self.__bgrp=''
```

```
    def Input(self):
        self.__name=raw_input("Enter the name of the donor: ")
        self.__DOB=raw_input("Enter the date of birth: ")
        self.__address=raw_input("Enter the address: ")
        self.__pno=int(raw_input("Enter phone number: "))
        self.__bgrp=raw_input("Enter the blood group: ")
        a=open(self.__name,'w')
        a.close()

    def Append(self):
        name="Donor name: "+str(self.__name)+'\n'
        dob="Date of birth: "+str(self.__DOB)+'\n'
        add="Address :"+str(self.__address)+'\n'
        pn="Phone number: "+str(self.__pno)+'\n'
        bg="Blood group: "+str(self.__bgrp)+'\n'
        l=[name,dob,add,pn,bg]
        a=open(self.__name,'a+')
        a.writelines(l)
        a.close()

    def Show(self):
        a=open(self.__name,'r')
        string=a.read()
        print
        print string
        a.close()

    @staticmethod
    def Search():
        bgp=raw_input("Enter the blood group for details: ")
        for x in l:
            if x.__bgrp==bgp:
                print "Name : ",x.__name
                print "Address: ",x.__address


no=int(raw_input("Enter the number of donors: "))
print
l=[]
for y in range (no):
    obj=Donor()
    obj.Input()
    print
    obj.Append()
    l.append(obj)
for x in l:
    print "DETAILS"
    x.Show()
    print "-------"

ans='y'
while ans=='y':
    ans=raw_input("Do you want to search details?(y/n) ")
    if ans=='y':
        Donor.Search()
```

Output:



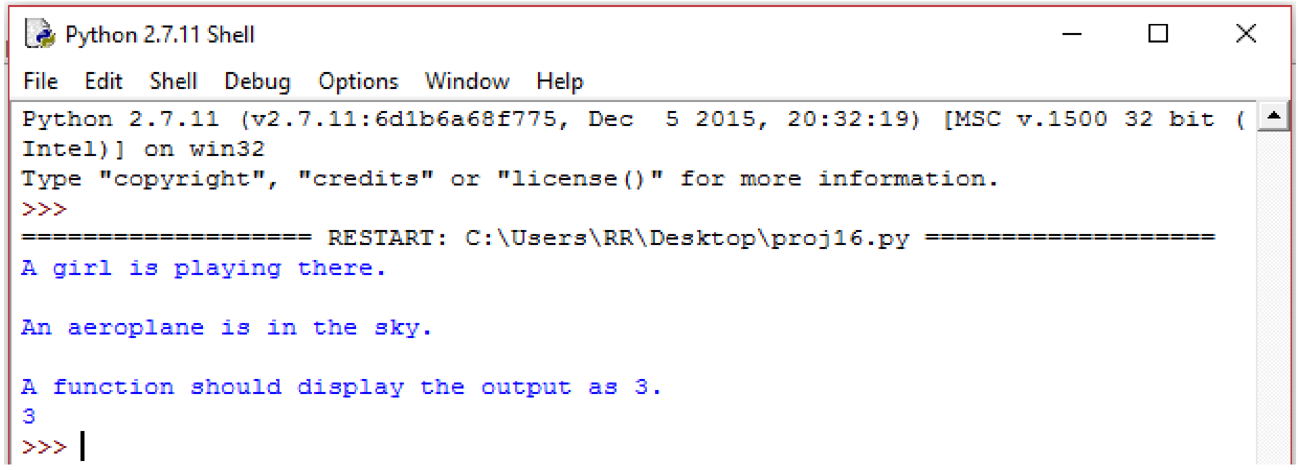
```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
================== RESTART: C:\Python27\Projects\proj15.py ==================
Enter the number of donors: 2

Enter the name of the donor: Dan
Enter the date of birth: 060988
Enter the address: somewhere
Enter phone number: 9897845673
Enter the blood group: O+ve

Enter the name of the donor: Wheldon
Enter the date of birth: 060988
Enter the address: Car
Enter phone number: 9567863210
Enter the blood group: A+ve

DETAILS

Donor name: Dan
Date of birth: 060988
Address :somewhere
Phone number: 9897845673
Blood group: O+ve

-------
DETAILS

Donor name: Wheldon
Date of birth: 060988
Address :Car
Phone number: 9567863210
Blood group: A+ve

-------
Do you want to search details?(y/n)
```

16.Write a program in Python to count and display the number of lines not starting with alphabet 'A' present in a text file “STORY.TXT” Example: If the file “STORY.TXT” contains the following lines,
The rose is red.
A girl is playing there.
There is a playground.
An aero plane is in the sky.
Numbers are not allowed in the password
The function should display the output as 3.

Code:

```
#File will have to be created on desktop
f=open('story.txt','r')
count=0
for i in f.readlines():
   if i[0]=='A':
      print i
      count=count+1
print count
```

Output:



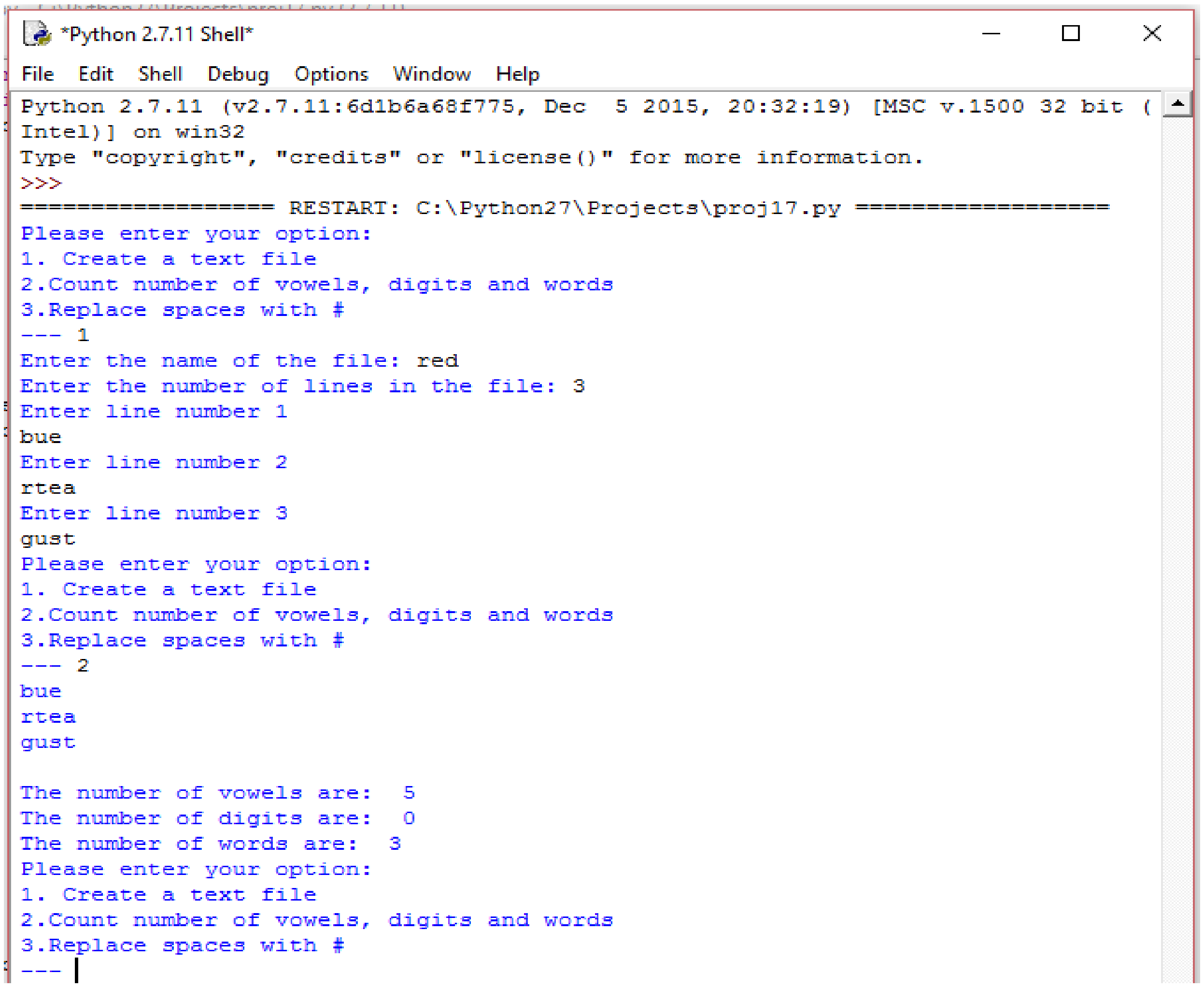
```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
=================== RESTART: C:\Users\RR\Desktop\proj16.py ===================
A girl is playing there.

An aeroplane is in the sky.

A function should display the output as 3.
3
>>> 
```

17.Write a menu driven program to:
a) Create a text file
b) Count number of vowels, digits and words
c)Create another file using the original which will contain the text after replacing all the blanks spaces with #.

Code:

```
while True:
   ch=int(raw_input("Please enter your option:"+'\n'+"1. Create a text file"+'\n'+'2.Count number of vowels, digits and words'+'\n'+'3.Replace spaces with #'+'\n'+'--- '))
   if ch==1:
      name=raw_input("Enter the name of the file: ")
      user=open(name,'w+')
      l=[]
      no=int(raw_input("Enter the number of lines in the file: "))
      for x in range (no):
         string=raw_input("Enter line number "+str(x+1) +'\n')
         l.append(string)
         l.append('\n')
      user.writelines(l)
      user.close()
   vowels=['a','e','i','o','u']
   if ch==2:
      vcount=0
      dcount=0
      wcount=0
      count=0
      user=open(name,'r')
      rd=user.read()
      print rd
      for x in rd:
         if x in vowels:
            vcount+=1
         elif x.isdigit():
            dcount+=1
         else:
            if x=='\n' or x==' ':
                count+=1
            wcount=count
      user.close()
      print "The number of vowels are: ",vcount
      print "The number of digits are: ",dcount
      print "The number of words are: ", wcount
   if ch==3:
      user=open(name,'a+')
      rd=user.readlines()
      li3=[]
      for x in rd:
         li=list(x)
         print
         for y in range(0,len(li)):
            if li[y]==' ':
                li[y]='#'
         st=''
         for a in li:
            st+=str(a)
         li3.append(st)
      user.close()
      user=open(name,'w+')
      user.writelines(li3)
      user.close()
```

Output:



Python 2.7.11 Shell

File Edit Shell Debug Options Window Help

```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
================== RESTART: C:\Python27\Projects\proj17.py ==================
Please enter your option:
1. Create a text file
2.Count number of vowels, digits and words
3.Replace spaces with #
--- 1
Enter the name of the file: red
Enter the number of lines in the file: 3
Enter line number 1
bue
Enter line number 2
rtea
Enter line number 3
gust
Please enter your option:
1. Create a text file
2.Count number of vowels, digits and words
3.Replace spaces with #
--- 2
bue
rtea
gust

The number of vowels are:   5
The number of digits are:   0
The number of words are:   3
Please enter your option:
1. Create a text file
2.Count number of vowels, digits and words
3.Replace spaces with #
--- 
```

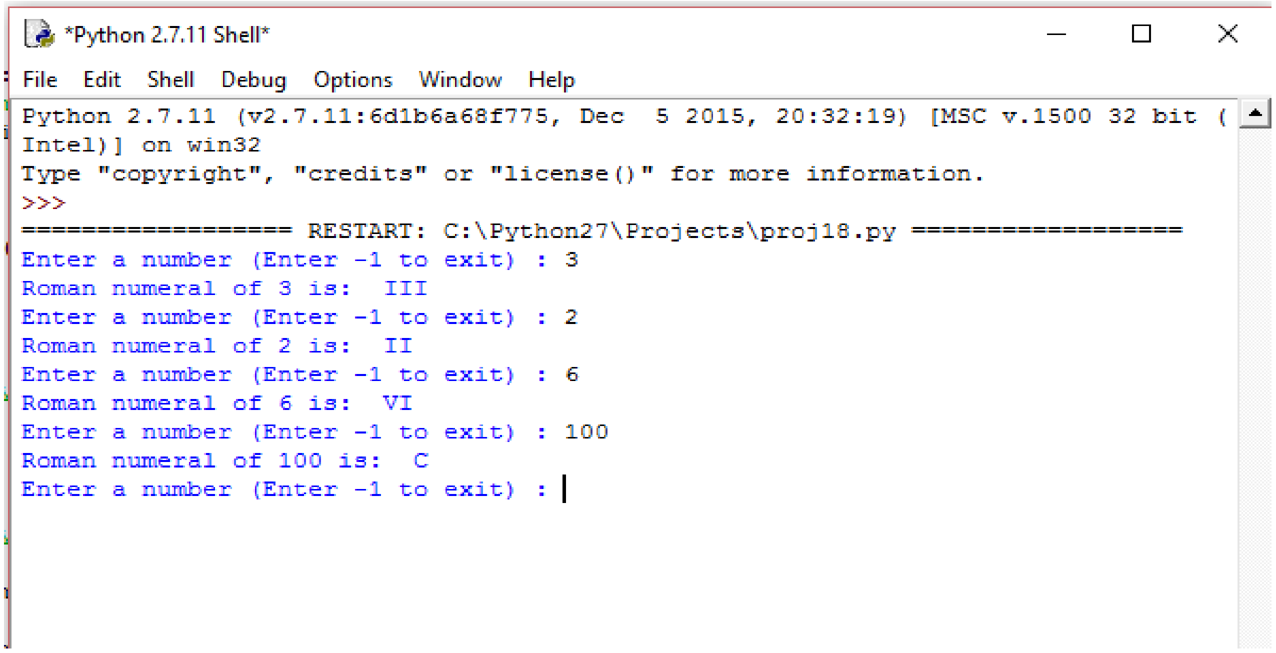
18.Write a program that stores the decimal numbers and equivalent roman numbers in the form of a dictionary in a file. Next, read the file, load the dictionary and ask the user to enter a numeral and print its equivalent roman number.

Code:

```
import pickle

roman={1000:"M",900:"CM",500:"D",400:"CD",100:"C",90:"XC",50:"L",40:"XL",10:"X", 9:"IX",5:"V",
4:"IV",1:"I"}
file1=open("RomanNum.log","wb")
pickle.dump(roman,file1)
file1.close()

while True:
  n=int(raw_input("Enter a number (Enter -1 to exit) : "))
  s=list(str(n))
  x=list()
```

```
    p=len(s)
    if p>4:
        print "Sorry, you can enter a maximum of 4 digits. Enter again"
        continue
    if n==-1:
        break
    if n>3999:
        print "Sorry you cannot represent any number greater than 3999 with roman numberals. Enter
another number"
        continue
        x.append(roman[1000]*int(s[-4]))
    if n/1000!=0:
        x.append(roman[1000]*int(s[0]))
    if n/100!=0:
        if s[-3]==9:
            x.append(roman[900])
        elif int(s[-3])>5 and int(s[-3])<9:
            x.append(roman[500]+roman[100]*(int(s[-3])-5))
        elif int(s[-3])==5:
            x.append(roman[500])
        elif int(s[-3])==4:
            x.append(roman[400])
        elif int(s[-3])<4:
            x.append(roman[100]*(int(s[-3])))

    if n/10!=0:
        if s[-2]=="9":
            x.append(roman[90])
        elif int(s[-2])>5 and int(s[-2])<9:
            x.append(roman[50]+roman[10]*(int(s[-2])-5))
        elif s[-2]=="5":
            x.append(roman[50])
        elif s[-2]=="4":
            x.append(roman[40])
        elif int(s[-2])<4:
            x.append(roman[10]*(int(s[-2])))

    if s[-1]=="9":
            x.append(roman[9])
    elif int(s[-1])>5 and int(s[-1])<9:
        x.append(roman[5]+roman[1]*(int(s[-1])-5))
    elif s[-1]=="5":
        x.append(roman[5])
    elif s[-1]=="4":
        x.append(roman[4])
    elif int(s[-1])<4:
        x.append(roman[1]*(int(s[-1])))

    print "Roman numeral of",n,"is: ",
    c=""
    for i in x:
        if i=="" or i==" ":
            continue
        c+=i
    print c
```

Output:



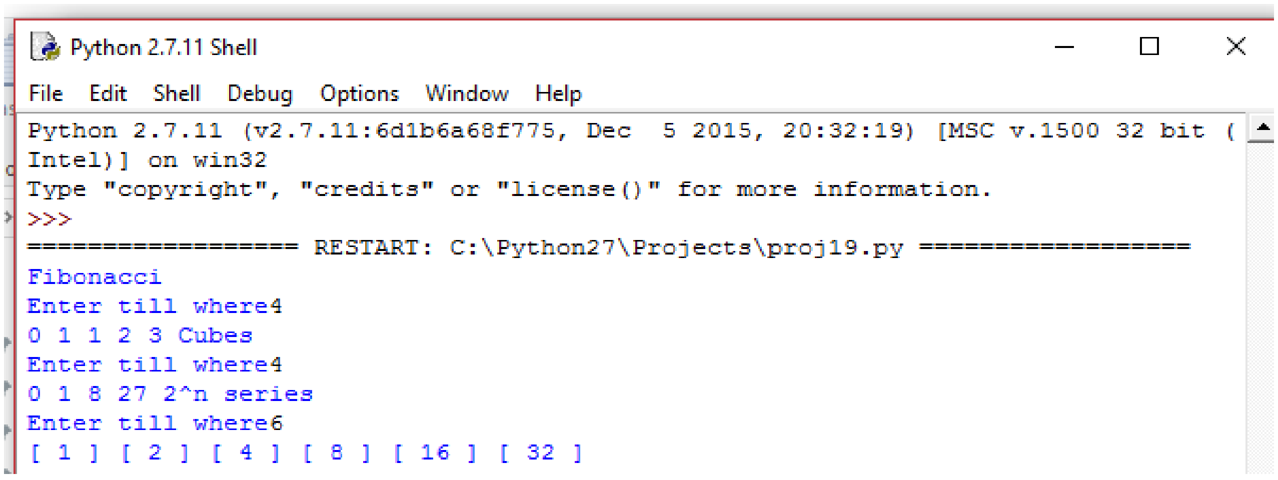
```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
================== RESTART: C:\Python27\Projects\proj18.py ==================
Enter a number (Enter -1 to exit) : 3
Roman numeral of 3 is:  III
Enter a number (Enter -1 to exit) : 2
Roman numeral of 2 is:  II
Enter a number (Enter -1 to exit) : 6
Roman numeral of 6 is:  VI
Enter a number (Enter -1 to exit) : 100
Roman numeral of 100 is:  C
Enter a number (Enter -1 to exit) : 
```

19.Write a generator program in Python to find:
a) Cube of a number
b) Fibonacci Series
c)2^n Series

Code:

```
def fibonacci(max):
    a,b=0,1
    while a<max:
        yield a
        a,b=b,a+b
n=int(raw_input("Enter till where"))
for i in fibonacci(n):
    print i,

def cube(max):
    a=0
    while a<max:
        yield a**3
        a=a+1
n=int(raw_input("Enter till where"))
for i in cube(n):
    print i,

def series(max):
    a=0
    while a<max:
        yield 2**a
        a=a+1
n=int(raw_input("Enter till where"))
for i in series(n):
    print "[",i,"]",
```

Output:



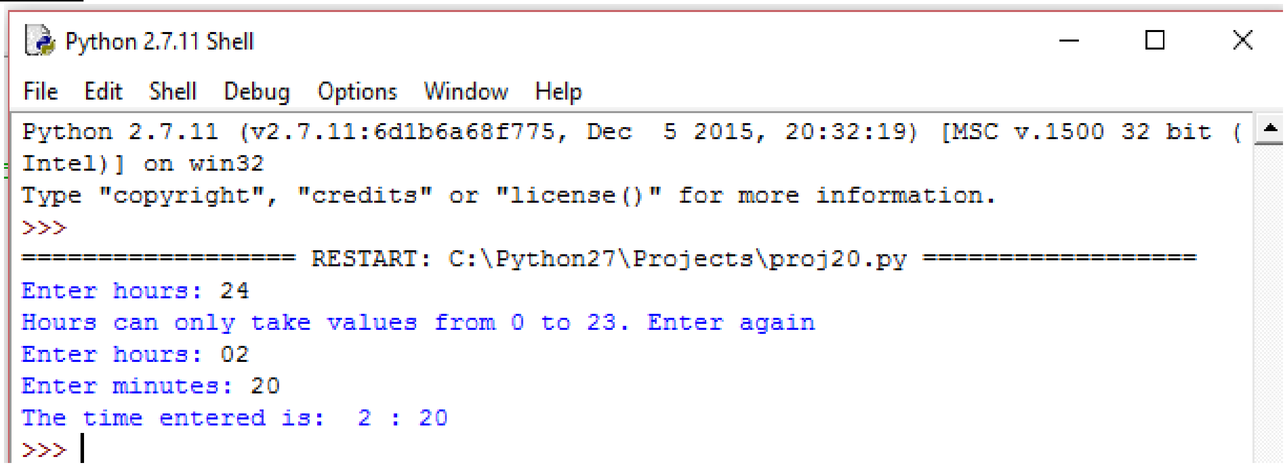
```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
================== RESTART: C:\Python27\Projects\proj19.py ==================
Fibonacci
Enter till where4
0 1 1 2 3 Cubes
Enter till where4
0 1 8 27 2^n series
Enter till where6
[ 1 ] [ 2 ] [ 4 ] [ 8 ] [ 16 ] [ 32 ]
```

20.Write a function to read a Time class object storing hours and minutes. Raise a user-defined error if the values other than 0.23 is entered for hours and other than 0.59 is entered for minutes.

Code:

```
class Time_error(Exception):
    pass
hour=Time_error("Hours can only take values from 0 to 23.")
minute=Time_error("Minutes can only take values from 0 to 59.")


class Time:
    def __init__(self):
        self.hour=0
        self.min=0

    def get(self):
        while True:
            try:
                self.hour=int(raw_input("Enter hours: "))
                if self.hour >23:
                    raise hour
                self.min=int(raw_input("Enter minutes: "))
                if self.min>59:
                    raise minute
                break
            except Time_error,e:
                print e.message,"Enter again"
                continue


    def display(self):
        print "The time entered is: ",self.hour,":",self.min

x=Time()
x.get()
x.display()
```

Output:



```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:32:19) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
================== RESTART: C:\Python27\Projects\proj20.py ==================
Enter hours: 24
Hours can only take values from 0 to 23. Enter again
Enter hours: 02
Enter minutes: 20
The time entered is:  2 : 20
>>> 
```