RAYALASEEMA UNIVERSITY

KURNOOL – 518 007

Department of Computer Science

CERTIFICATE

MCA –V and M.Sc (Data Science) –III Semester

Subject: PYTHON PROGRAMMING LAB

Register No:

Practical No:

This is to certify that Mr./Mrs./Kumari. has successfully completed his/her III Semester Practical assignment during the academic year 2019-20 in the computer Laboratories of the Rayalaseema University.

No. of Problems assigned: □ No. of Problems solved: □

Subject in charge **Coordinator**

(Dr. G. RAVI KUMAR)

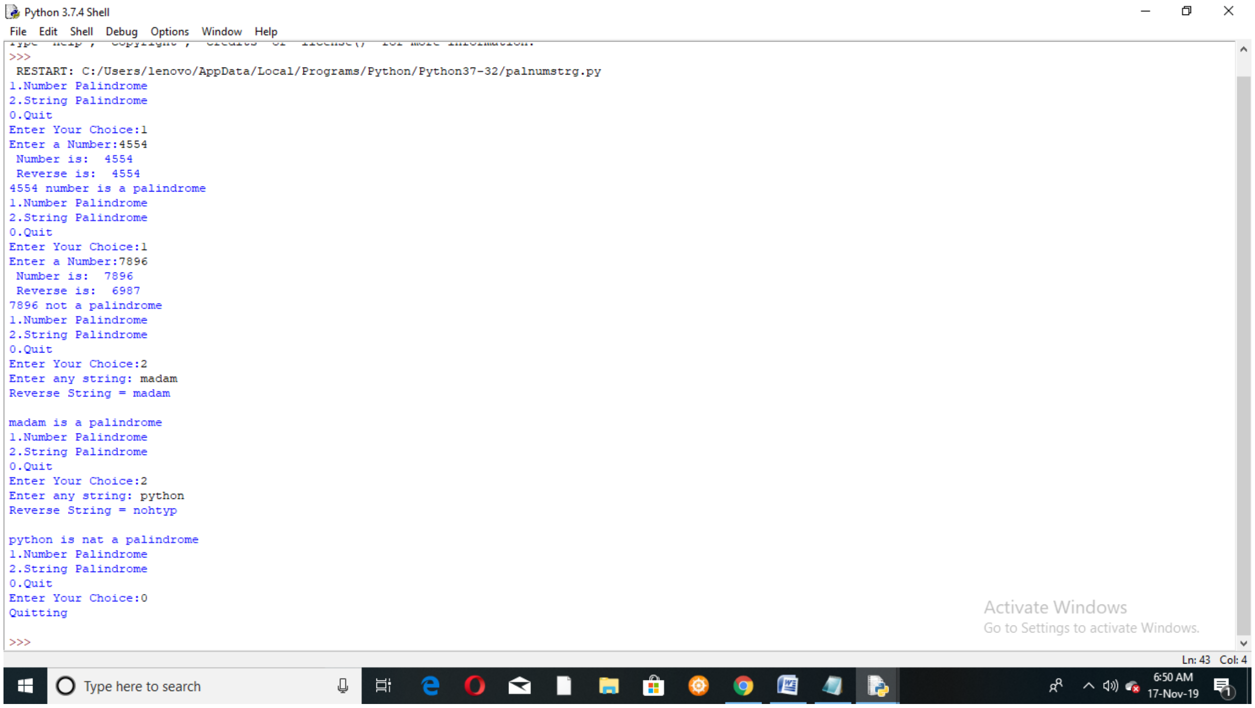
Examiners

1.

1. Write a Program in python to check whether the given number and string are palindrome or not

```
choice = 1
while choice !=0:
    print("1.Number Palindrome")
    print("2.String Palindrome")
    print("0.Quit")
    choice = int(input("Enter Your Choice:"))
    if choice == 1:
        num=int(input("Enter a Number:"))
        rev = 0
        n=num
        while num > 0:
            rem = num % 10
            rev = rev * 10 + rem
            num = num // 10
        print(' Number is: ', n)
        print(' Reverse is: ', rev)
        if n == rev:
            print (n,'number is a palindrome')
        else:
             print (n,'not a palindrome')
    elif choice == 2:
        string = input("Enter any string: ")
        revstring = string[::-1]
        print("Reverse String =",revstring,"\n")
        if(string==revstring):
            print(string,"is a palindrome")
        else:
            print(string,"is nat a palindrome")
    elif choice == 0:
        print("Quitting")
    else:
         print("Invalid Choice")
print()
```

OUTPUT

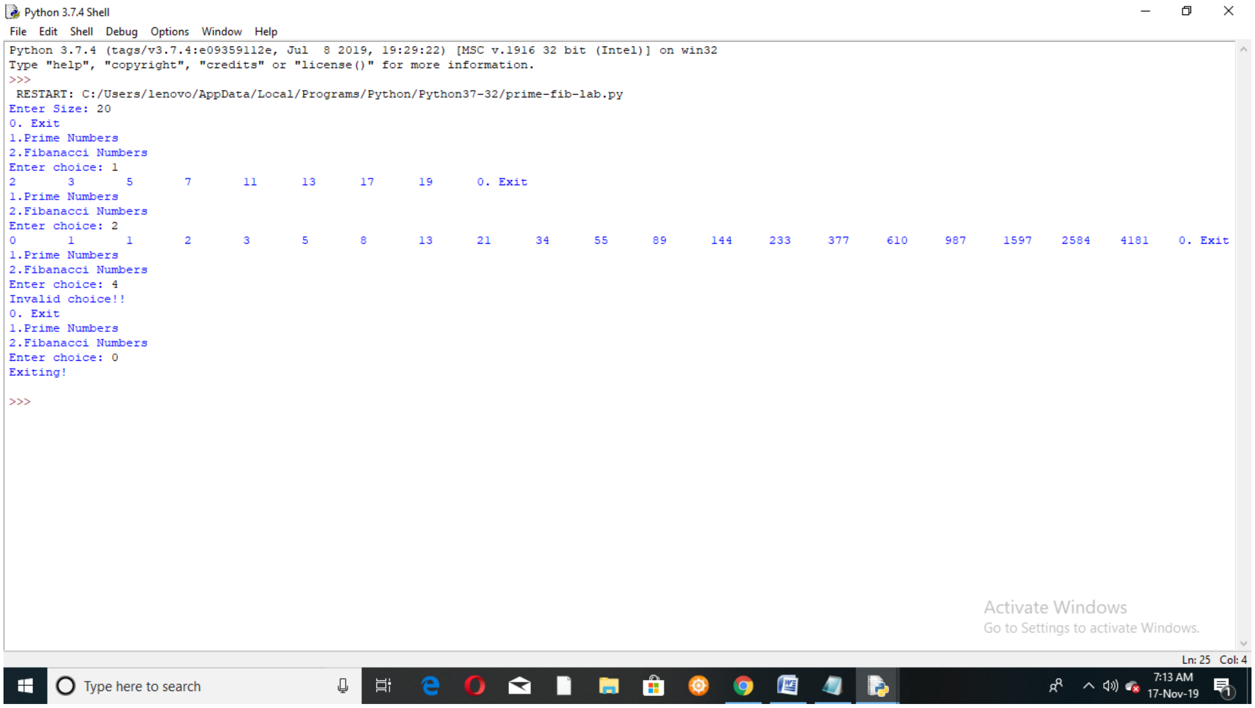


```
Type "help", "copyright", "credits" or "license()" for more information.
>>>
 RESTART: C:/Users/lenovo/AppData/Local/Programs/Python/Python37-32/palnumstrg.py
1.Number Palindrome
2.String Palindrome
0.Quit
Enter Your Choice:1
Enter a Number:4554
 Number is:  4554
 Reverse is:  4554
4554 number is a palindrome
1.Number Palindrome
2.String Palindrome
0.Quit
Enter Your Choice:1
Enter a Number:7896
 Number is:  7896
 Reverse is:  6987
7896 not a palindrome
1.Number Palindrome
2.String Palindrome
0.Quit
Enter Your Choice:2
Enter any string: madam
Reverse String = madam

madam is a palindrome
1.Number Palindrome
2.String Palindrome
0.Quit
Enter Your Choice:2
Enter any string: python
Reverse String = nohtyp

python is nat a palindrome
1.Number Palindrome
2.String Palindrome
0.Quit
Enter Your Choice:0
Quitting

>>>
```

2. **Write a Program in Python to generate N Prime and Fibonacci Numbers?**

```
n=int(input("Enter Size: "))
choice=1
while choice!=0:
    print("0. Exit")
    print("1.Prime Numbers")
    print("2.Fibanacci Numbers")
    choice=int(input("Enter choice: "))
    if choice==1:
       i=1
       x = n
       for k in range (1, (x+1), 1):
          c=0;
          for j in range (1, (i+1), 1):
             a = i%j
             if (a==0):
                c = c+1
          if (c==2):
             print(i,end="\t")
          else:
             k = k-1
          i=i+1
    elif choice==2:
       a = 0
       b = 1
       count = 2
       print(a,end="\t")
       print(b,end="\t")
       while count < n:
          c = a + b
          print(c,end="\t")
          a = b
          b = c
          count = count + 1
    elif choice==0:
        print("Exiting!")
    else:
       print("Invalid choice!!")
    print()
```

OUTPUT

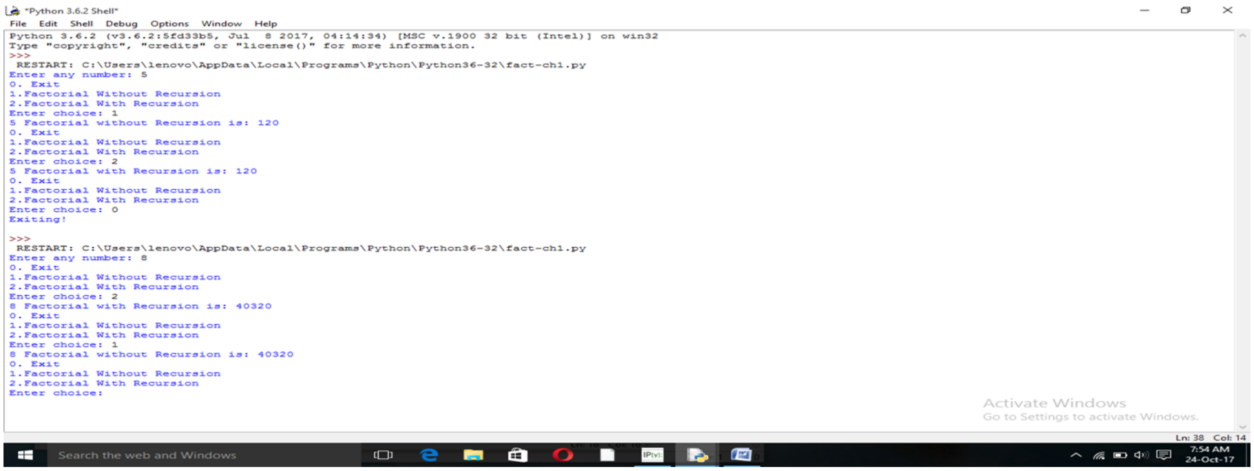


```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
 RESTART: C:/Users/lenovo/AppData/Local/Programs/Python/Python37-32/prime-fib-lab.py
Enter Size: 20
0. Exit
1.Prime Numbers
2.Fibanacci Numbers
Enter choice: 1
2       3       5       7       11      13      17      19      0. Exit
1.Prime Numbers
2.Fibanacci Numbers
Enter choice: 2
0       1       1       2       3       5       8       13      21      34      55      89      144     233     377     610     987     1597    2584    4181    0. Exit
1.Prime Numbers
2.Fibanacci Numbers
Enter choice: 4
Invalid choice!!
0. Exit
1.Prime Numbers
2.Fibanacci Numbers
Enter choice: 0
Exiting!

>>> 
```

3. Write a Python program to find the Factorial of a given number with and without Recursion

```
def fact1(n):

        fact=1

        for i in range(1,n+1):

             fact=fact*i

        return fact

def factorial(n):

        if(n <= 1):

            return 1

        else:

            return(n*factorial(n-1))

n=int(input("Enter any number: "))

choice=1

while choice!=0:

      print("0. Exit")

      print("1.Factorial Without Recursion")

      print("2.Factorial With Recursion")

      choice=int(input("Enter choice: "))

      if choice==1:

        print(n,"Factorial without Recursion is:",fact1(n))

      elif choice==2:

        print(n,"Factorial with Recursion is:",factorial(n))

      elif choice==0:
            print("Exiting!")
        else:
            print("Invalid choice!!")

print()
```

OUTPUT



Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Activate Windows
Go to Settings to activate Windows.
Ln: 38 Col: 14
Search the web and Windows
7:54 AM
24-Oct-17

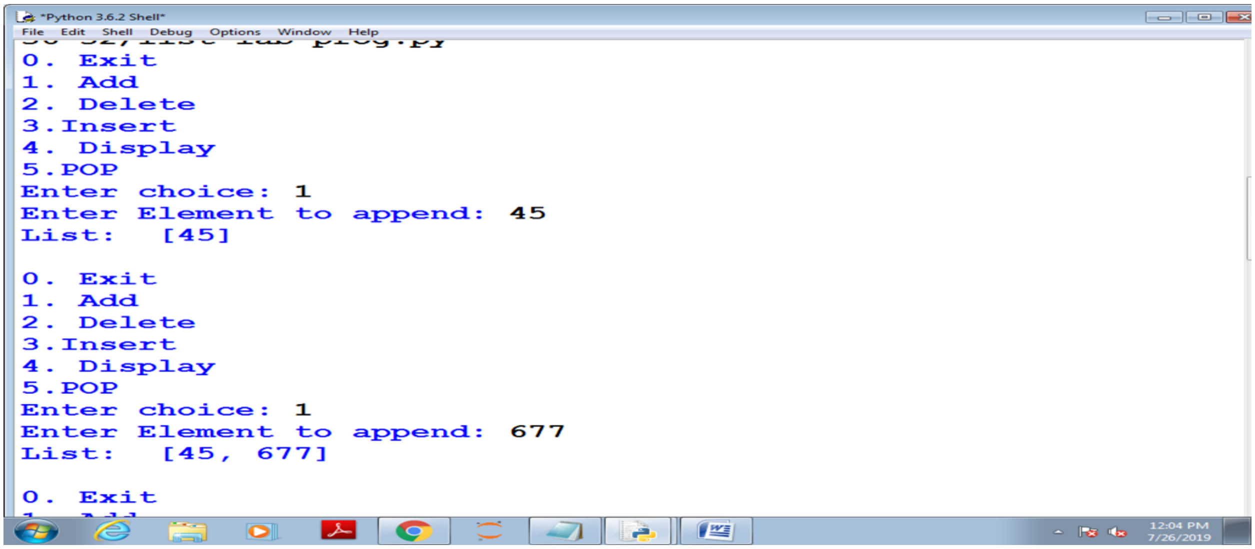
```
Python 3.6.2 (v3.6.2:5fd33b5, Jul  8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
 RESTART: C:\Users\lenovo\AppData\Local\Programs\Python\Python36-32\fact-ch1.py 
Enter any number: 5
0. Exit
1.Factorial Without Recursion
2.Factorial With Recursion
Enter choice: 1
5 Factorial without Recursion is: 120
0. Exit
1.Factorial Without Recursion
2.Factorial With Recursion
Enter choice: 2
5 Factorial with Recursion is: 120
0. Exit
1.Factorial Without Recursion
2.Factorial With Recursion
Enter choice: 0
Exiting!

>>> 
 RESTART: C:\Users\lenovo\AppData\Local\Programs\Python\Python36-32\fact-ch1.py 
Enter any number: 8
0. Exit
1.Factorial Without Recursion
2.Factorial With Recursion
Enter choice: 2
8 Factorial with Recursion is: 40320
0. Exit
1.Factorial Without Recursion
2.Factorial With Recursion
Enter choice: 1
8 Factorial without Recursion is: 40320
0. Exit
1.Factorial Without Recursion
2.Factorial With Recursion
Enter choice: 
```

4. Writa a Program in Python to perform various List operations

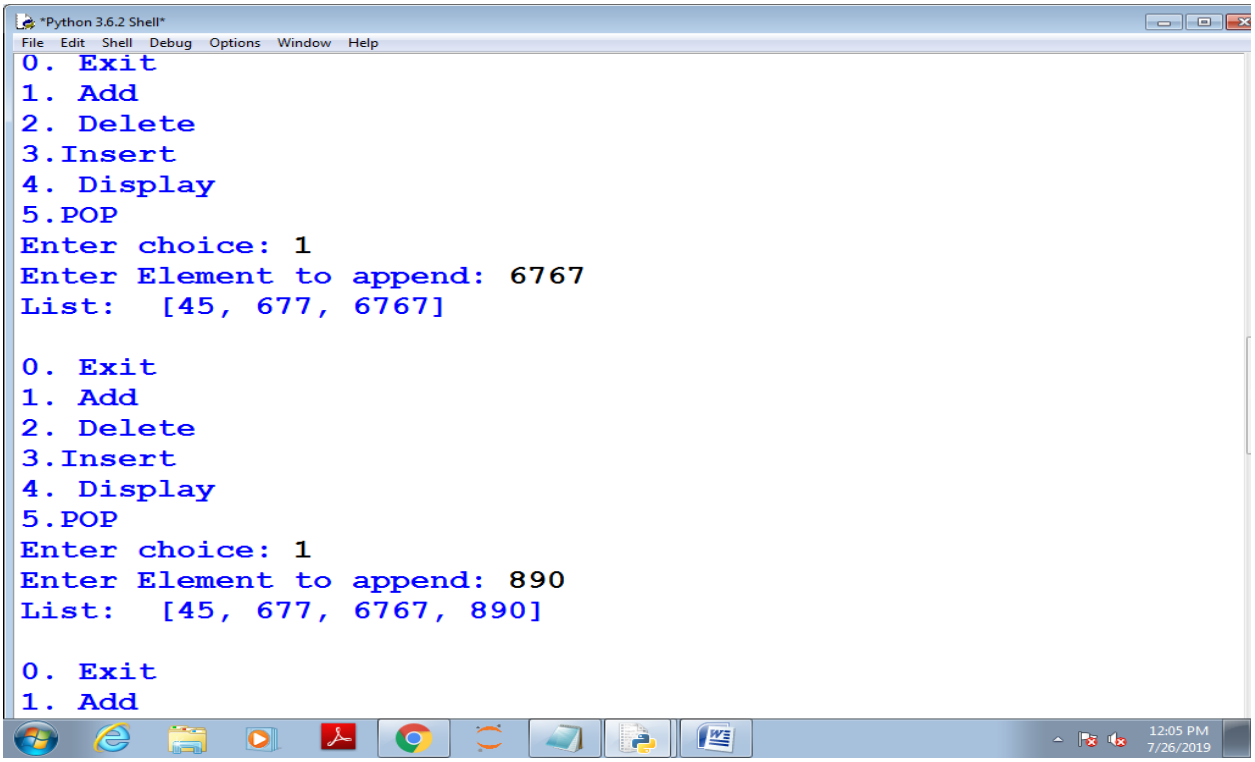
```
"""Write a Program in Python to demonstrate List operation"""

n=[]

def dis():

    return n

choice=1

while choice!=0:

    print("0. Exit")

    print("1. Add")

    print("2. Delete")

    print("3.Insert")

    print("4. Display")

    print("5.POP")

    choice=int(input("Enter choice: "))

    if choice==1:

        a = int(input("Enter Element to append: "))

        n.append(a)

        print("List: ",dis())

    elif choice==2:

        if n == []:

            print("No elements in the List")

        else:

           b=eval(input("Enter number to remove: "))

           n.remove(b)
```

```
            print("List: ",dis())
    elif choice==3:
        pos = eval(input("Enter a position to insert"))
        value = eval(input("Enter a Number to insert"))
        n.insert(pos,value)
        print("List: ",dis())
    elif choice==4:
        print("List: ",dis())
    elif choice == 5:
        n.pop()
        print("List:",dis())
    elif choice==0:
        print("Exiting!")
    else:
        print("Invalid choice!!")
    print()
```

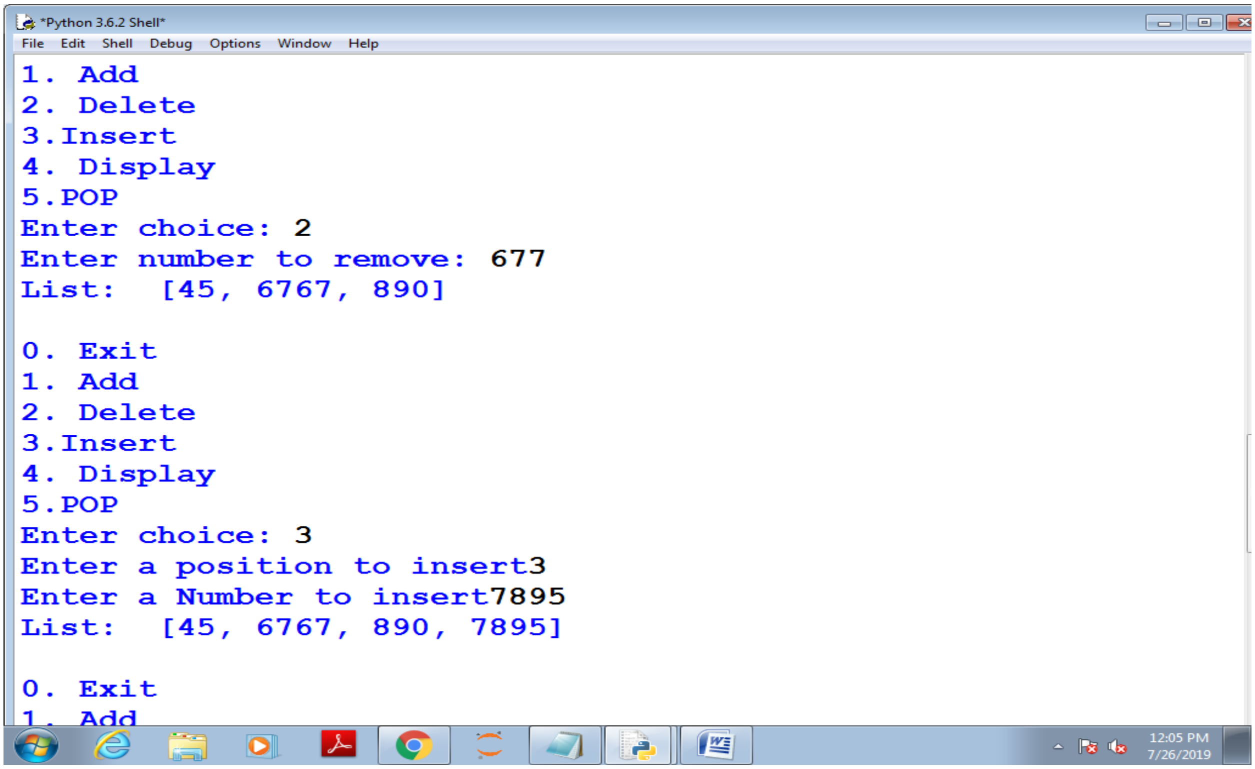
OUTPUT



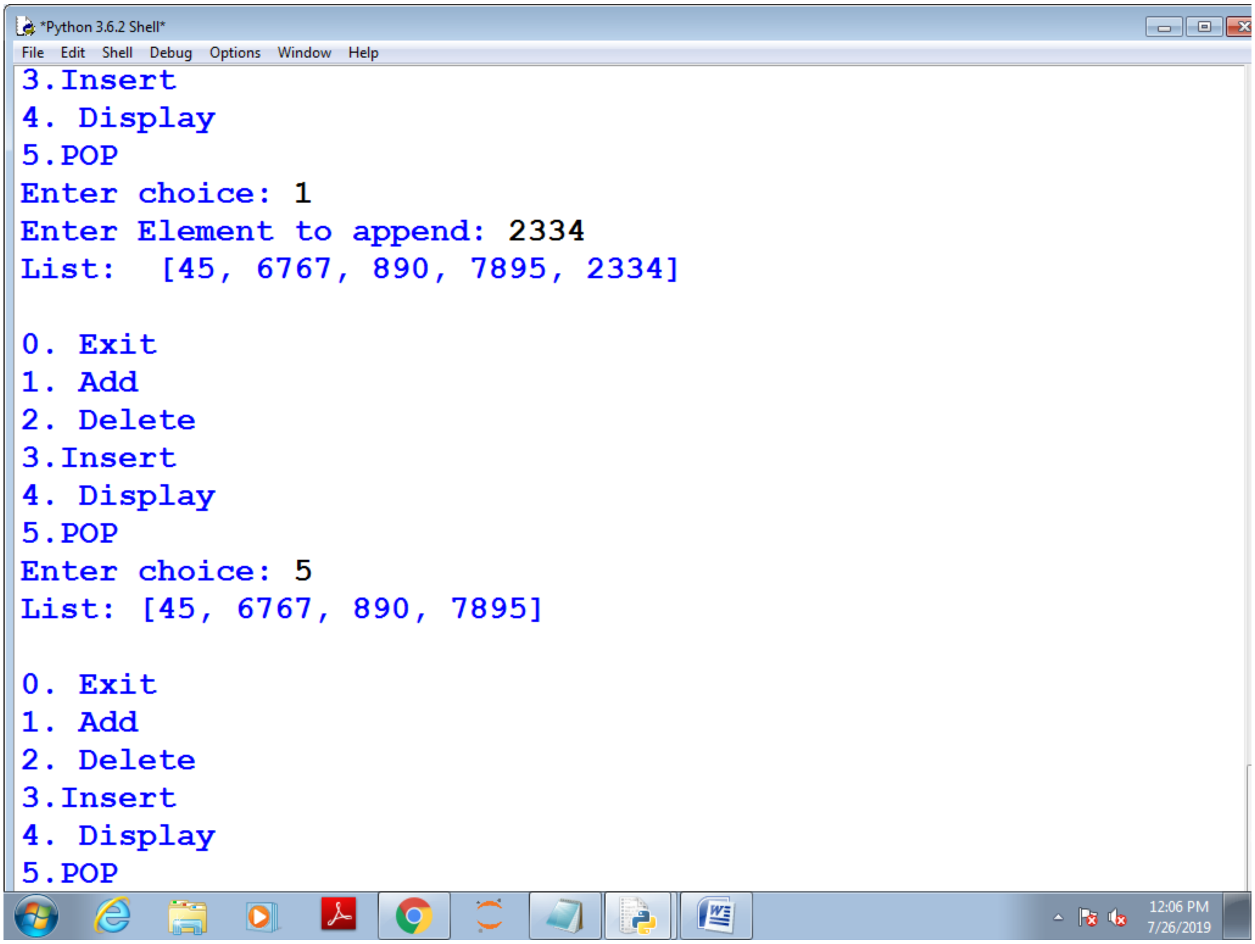
```
0.  Exit
1.  Add
2.  Delete
3.Insert
4.  Display
5.POP
Enter choice: 1
Enter Element to append: 45
List:   [45]

0.  Exit
1.  Add
2.  Delete
3.Insert
4.  Display
5.POP
Enter choice: 1
Enter Element to append: 677
List:   [45, 677]

0.  Exit
```



```
0. Exit
1. Add
2. Delete
3.Insert
4. Display
5.POP
Enter choice: 1
Enter Element to append: 6767
List:  [45, 677, 6767]

0. Exit
1. Add
2. Delete
3.Insert
4. Display
5.POP
Enter choice: 1
Enter Element to append: 890
List:  [45, 677, 6767, 890]

0. Exit
1. Add
```



```
1. Add
2. Delete
3.Insert
4. Display
5.POP
Enter choice: 2
Enter number to remove: 677
List:  [45, 6767, 890]

0. Exit
1. Add
2. Delete
3.Insert
4. Display
5.POP
Enter choice: 3
Enter a position to insert3
Enter a Number to insert7895
List:  [45, 6767, 890, 7895]

0. Exit
1. Add
```



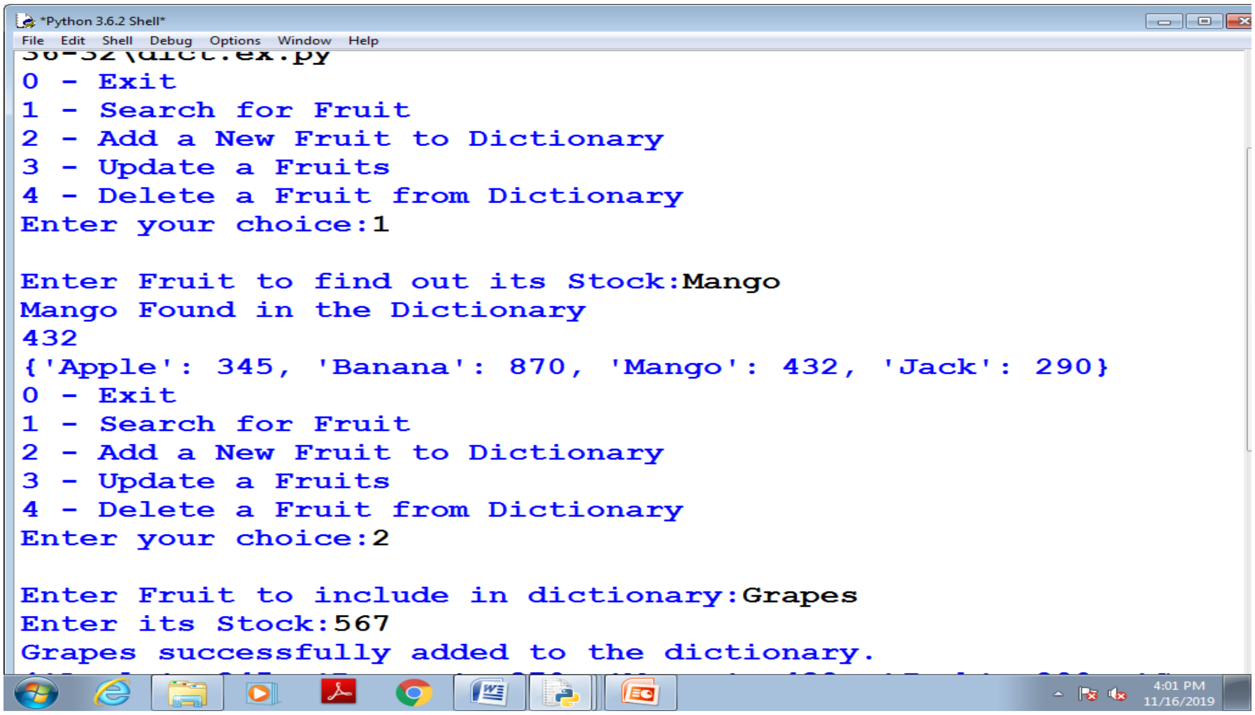
```
3.Insert
4. Display
5.POP
Enter choice: 1
Enter Element to append: 2334
List:  [45, 6767, 890, 7895, 2334]

0. Exit
1. Add
2. Delete
3.Insert
4. Display
5.POP
Enter choice: 5
List: [45, 6767, 890, 7895]

0. Exit
1. Add
2. Delete
3.Insert
4. Display
5.POP
```

5. Write a Program in Python to display fruit stock details using a Dictionary

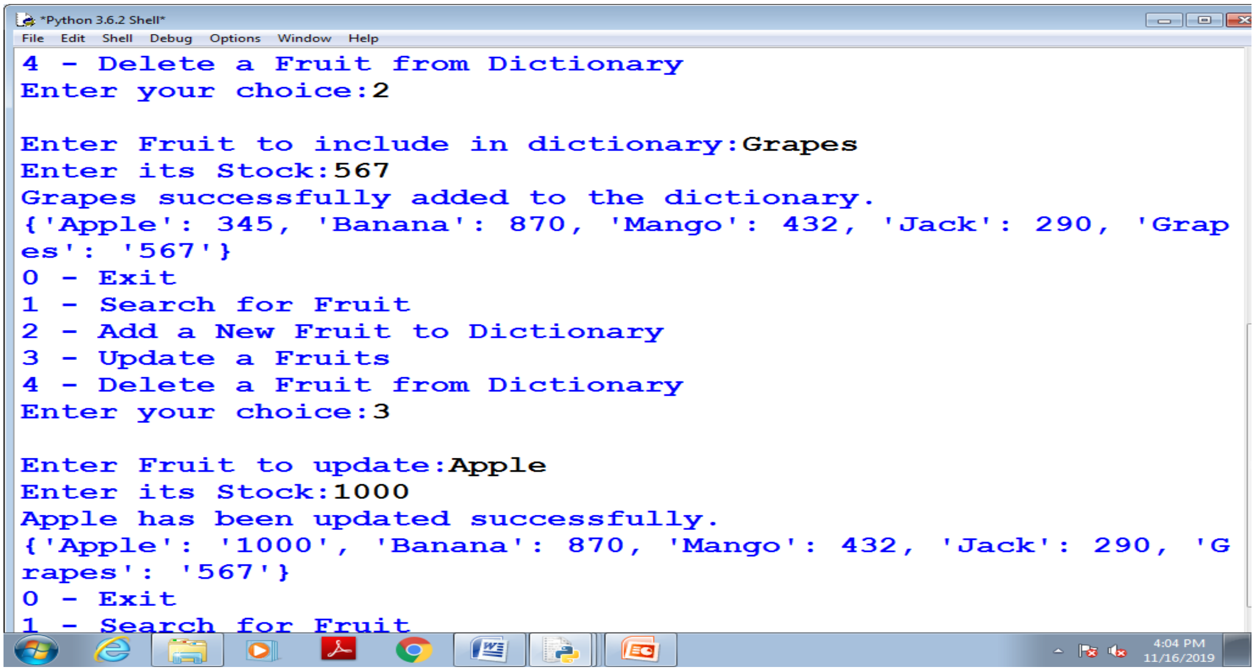
```
Fruits_dictionary = {"Apple": 345,"Banana": 870,"Mango":432,"Jack": 290}
choice = None
while choice != "0":
    print("0 - Exit")
    print("1 - Search for Fruit")
    print("2 - Add a New Fruit to Dictionary")
    print("3 - Update a Fruits")
    print("4 - Delete a Fruit from Dictionary")
    choice = input("Enter your choice:")
    if choice == "0":
        print("\nExiting....")
    elif choice == "1":
        fruit = input("\nEnter Fruit to find out its Stock:");
        if fruit in Fruits_dictionary:
            print(fruit,"Found in the Dictionary")
            fruit_stock = Fruits_dictionary[fruit]
            print(fruit_stock)
            print(Fruits_dictionary)
        else:
            print("not found!")
    elif choice == "2":
        fruit = input("\nEnter Fruit to include in dictionary:")
    if fruit not in Fruits_dictionary:
        fruit_stock = input("Enter its Stock:")
        Fruits_dictionary[fruit] = fruit_stock
        print(fruit,"successfully added to the dictionary.")
        print(Fruits_dictionary)
    else:
        print("already exists!")
elif choice == "3":
    fruit = input("\nEnter Fruit to update:")
    if fruit in Fruits_dictionary:
        fruit_stock = input("Enter its Stock:")
        Fruits_dictionary[fruit] = fruit_stock
        print(fruit,"has been updated successfully.")
        print(Fruits_dictionary)
    else:
        print("not found!")
elif choice == "4":
        fruit = input("\nEnter Fruit to delete it form dictionary:")
        if fruit in Fruits_dictionary:
            del Fruits_dictionary[fruit]
            print(fruit,"deleted from dictionary successfully.")
            print(Fruits_dictionary)
        else:
            print("not found!")
else:
```

```
print("\nChoice is not valid!")
```

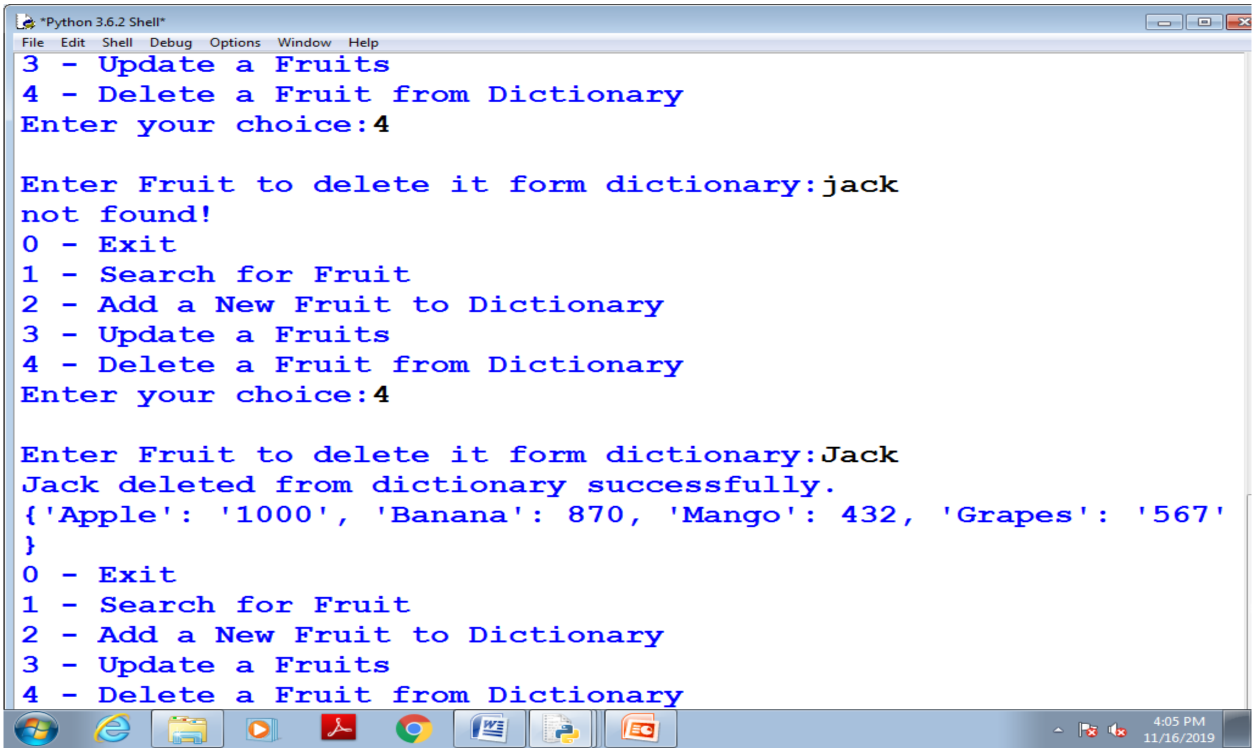
OUTPUT



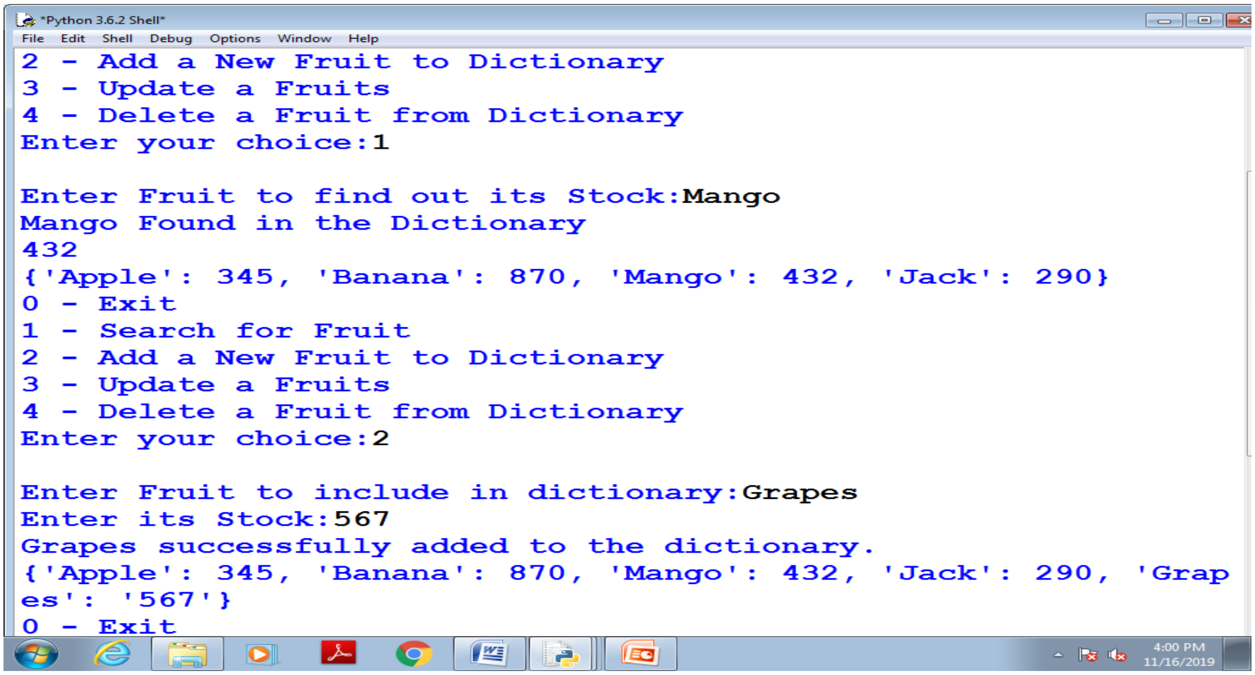
```
36-32\dict.ex.py
0 - Exit
1 - Search for Fruit
2 - Add a New Fruit to Dictionary
3 - Update a Fruits
4 - Delete a Fruit from Dictionary
Enter your choice:1

Enter Fruit to find out its Stock:Mango
Mango Found in the Dictionary
432
{'Apple': 345, 'Banana': 870, 'Mango': 432, 'Jack': 290}
0 - Exit
1 - Search for Fruit
2 - Add a New Fruit to Dictionary
3 - Update a Fruits
4 - Delete a Fruit from Dictionary
Enter your choice:2

Enter Fruit to include in dictionary:Grapes
Enter its Stock:567
Grapes successfully added to the dictionary.
```



```
4 - Delete a Fruit from Dictionary
Enter your choice:2

Enter Fruit to include in dictionary:Grapes
Enter its Stock:567
Grapes successfully added to the dictionary.
{'Apple': 345, 'Banana': 870, 'Mango': 432, 'Jack': 290, 'Grap
es': '567'}
0 - Exit
1 - Search for Fruit
2 - Add a New Fruit to Dictionary
3 - Update a Fruits
4 - Delete a Fruit from Dictionary
Enter your choice:3

Enter Fruit to update:Apple
Enter its Stock:1000
Apple has been updated successfully.
{'Apple': '1000', 'Banana': 870, 'Mango': 432, 'Jack': 290, 'G
rapes': '567'}
0 - Exit
1 - Search for Fruit
```



```
3 - Update a Fruits
4 - Delete a Fruit from Dictionary
Enter your choice:4

Enter Fruit to delete it form dictionary:jack
not found!
0 - Exit
1 - Search for Fruit
2 - Add a New Fruit to Dictionary
3 - Update a Fruits
4 - Delete a Fruit from Dictionary
Enter your choice:4

Enter Fruit to delete it form dictionary:Jack
Jack deleted from dictionary successfully.
{'Apple': '1000', 'Banana': 870, 'Mango': 432, 'Grapes': '567'
}
0 - Exit
1 - Search for Fruit
2 - Add a New Fruit to Dictionary
3 - Update a Fruits
4 - Delete a Fruit from Dictionary
```

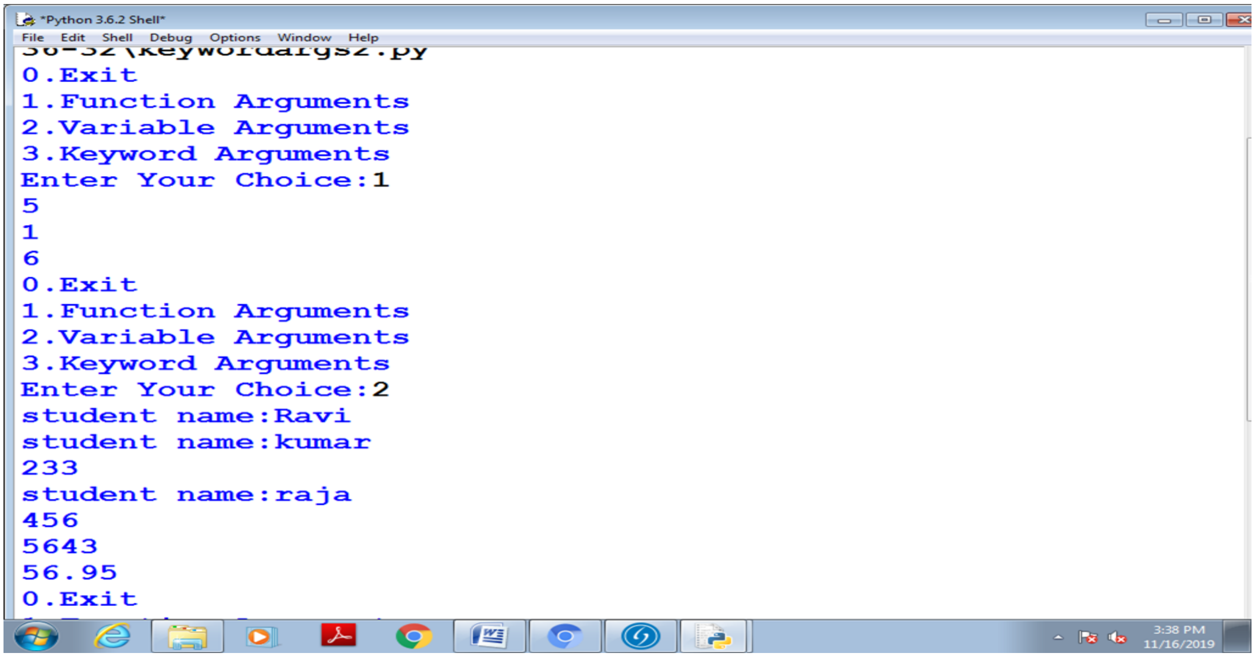


```
2 - Add a New Fruit to Dictionary
3 - Update a Fruits
4 - Delete a Fruit from Dictionary
Enter your choice:1

Enter Fruit to find out its Stock:Mango
Mango Found in the Dictionary
432
{'Apple': 345, 'Banana': 870, 'Mango': 432, 'Jack': 290}
0 - Exit
1 - Search for Fruit
2 - Add a New Fruit to Dictionary
3 - Update a Fruits
4 - Delete a Fruit from Dictionary
Enter your choice:2

Enter Fruit to include in dictionary:Grapes
Enter its Stock:567
Grapes successfully added to the dictionary.
{'Apple': 345, 'Banana': 870, 'Mango': 432, 'Jack': 290, 'Grap
es': '567'}
0 - Exit
```

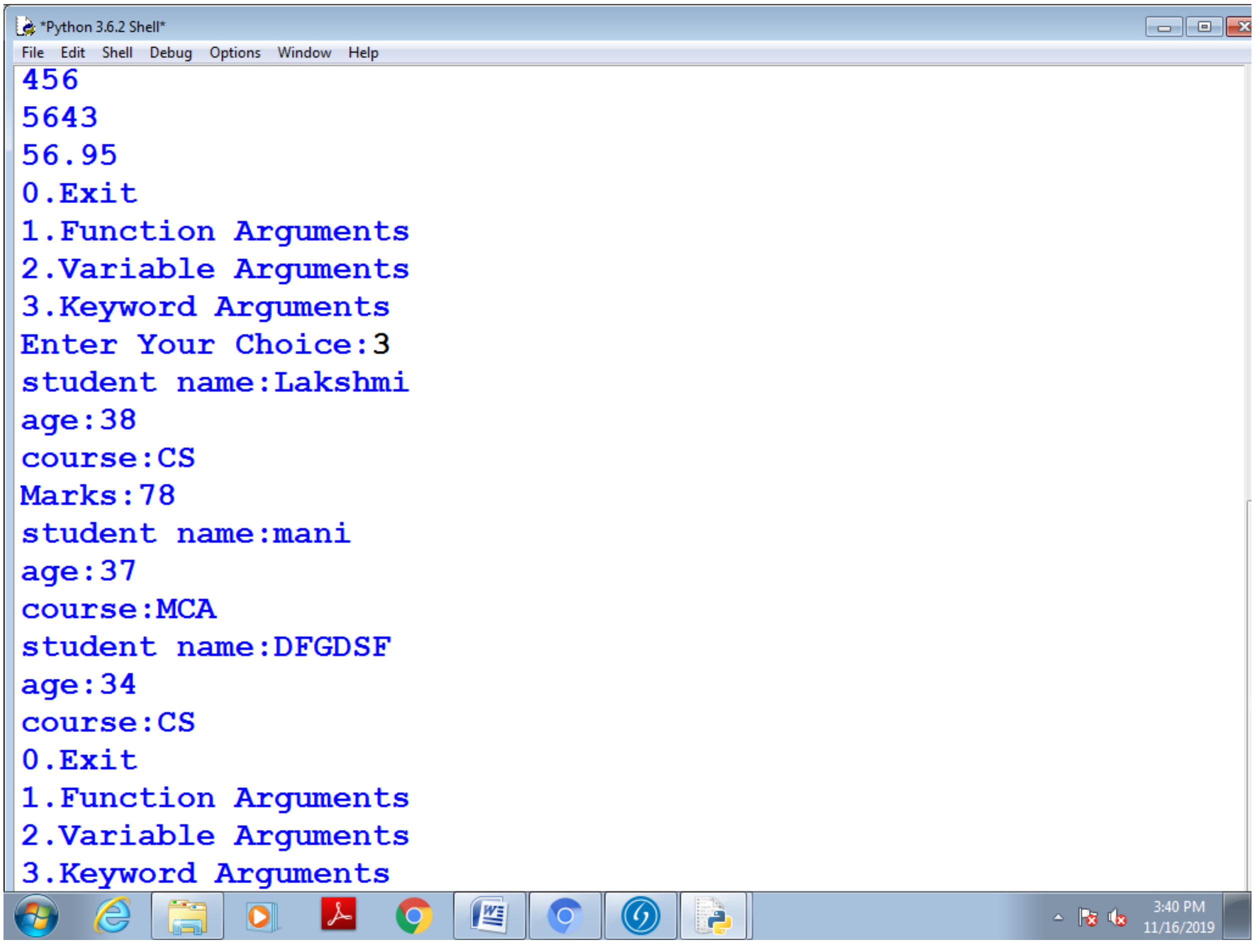
6. Write a Program in Python to Demonstrate a Variable No.of arguments and Keyword Arguments

```
def add(x, y):
   return x + y
def sub(x, y):
   return x - y
def my_apply(func, x, y):
   return func(x, y)
def student(name,*varargs):
   print('student name:' +name)
   for item in varargs:
      print(item)
def student1(name,**kwargs):
   print("student name:" +name)
   for key,value in kwargs.items():
      print(key + ":" + value)
choice=1
while choice !=0:
   print("0.Exit")
   print("1.Function Arguments")
   print("2.Variable Arguments")
   print("3.Keyword Arguments")
   choice=int(input("Enter Your Choice:"))
   if choice == 1:
      print(my_apply(add, 3, 2))

      print(my_apply(sub, 3, 2))

      print(my_apply(lambda x, y: x * y, 3, 2))

    elif choice ==2:

       student("Ravi")

       student("kumar",233)

       student("raja",456,5643,56.95)

    elif choice == 3:

       student1(name="Lakshmi",age='38',course='CS',Marks='78')

       student1(name="mani",age='37',course='MCA')

       student1(name="DFGDSF",age='34',course='CS')

    else:

      print("Exiting")

print()
```

OUTPUT



```
36-32\keywordargs2.py
0.Exit
1.Function Arguments
2.Variable Arguments
3.Keyword Arguments
Enter Your Choice:1
5
1
6
0.Exit
1.Function Arguments
2.Variable Arguments
3.Keyword Arguments
Enter Your Choice:2
student name:Ravi
student name:kumar
233
student name:raja
456
5643
56.95
0.Exit
```

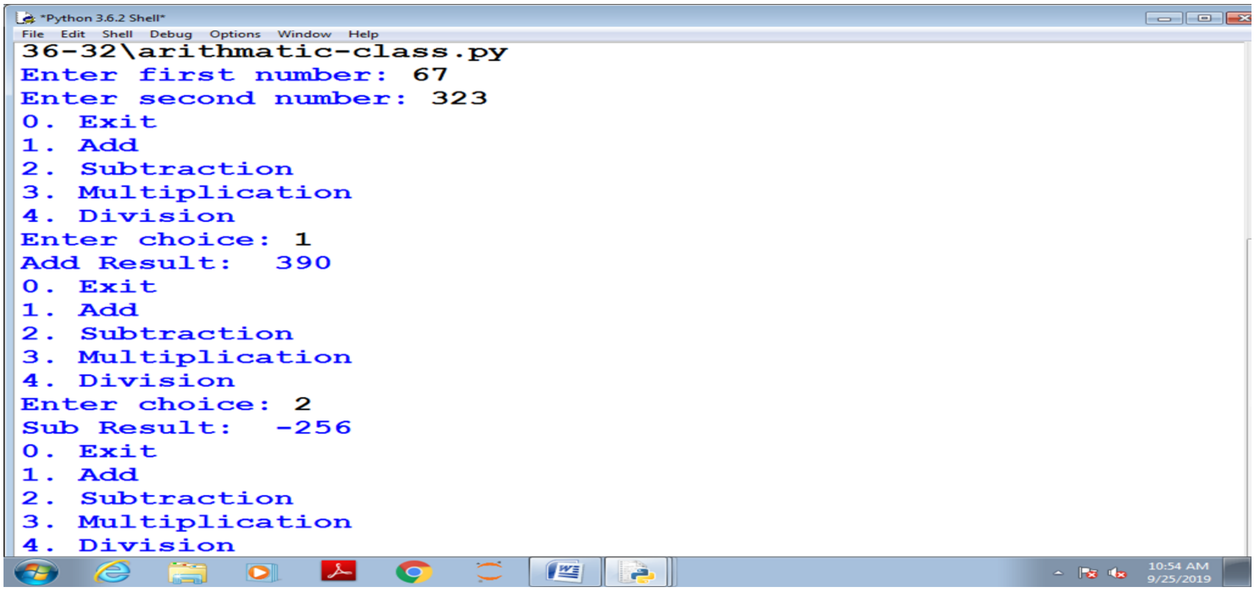


```
456
5643
56.95
0.Exit
1.Function Arguments
2.Variable Arguments
3.Keyword Arguments
Enter Your Choice:3
student name:Lakshmi
age:38
course:CS
Marks:78
student name:mani
age:37
course:MCA
student name:DFGDSF
age:34
course:CS
0.Exit
1.Function Arguments
2.Variable Arguments
3.Keyword Arguments
```

7. Write a Program in Python to perform arithmetical operations

```
class cal():
    def __init__(self,a,b):
        self.a=a
        self.b=b
    def add(self):
        return self.a+self.b
    def mul(self):
        return self.a*self.b
    def div(self):
        return self.a/self.b
    def sub(self):
        return self.a-self.b
a=int(input("Enter first number: "))
b=int(input("Enter second number: "))
obj=cal(a,b)
choice=1
while choice!=0:
    print("0. Exit")
    print("1. Add")
    print("2. Subtraction")
    print("3. Multiplication")
    print("4. Division")
    choice=int(input("Enter choice: "))
    if choice==1:

        print("Add Result: ",obj.add())

    elif choice==2:

        print("Sub Result: ",obj.sub())

    elif choice==3:

        print("Mul Result: ",obj.mul())

    elif choice==4:

        print("Div Result: ",round(obj.div(),2))

    elif choice==0:

        print("Exiting!")

    else:

        print("Invalid choice!!")


print()
```

OUTPUT

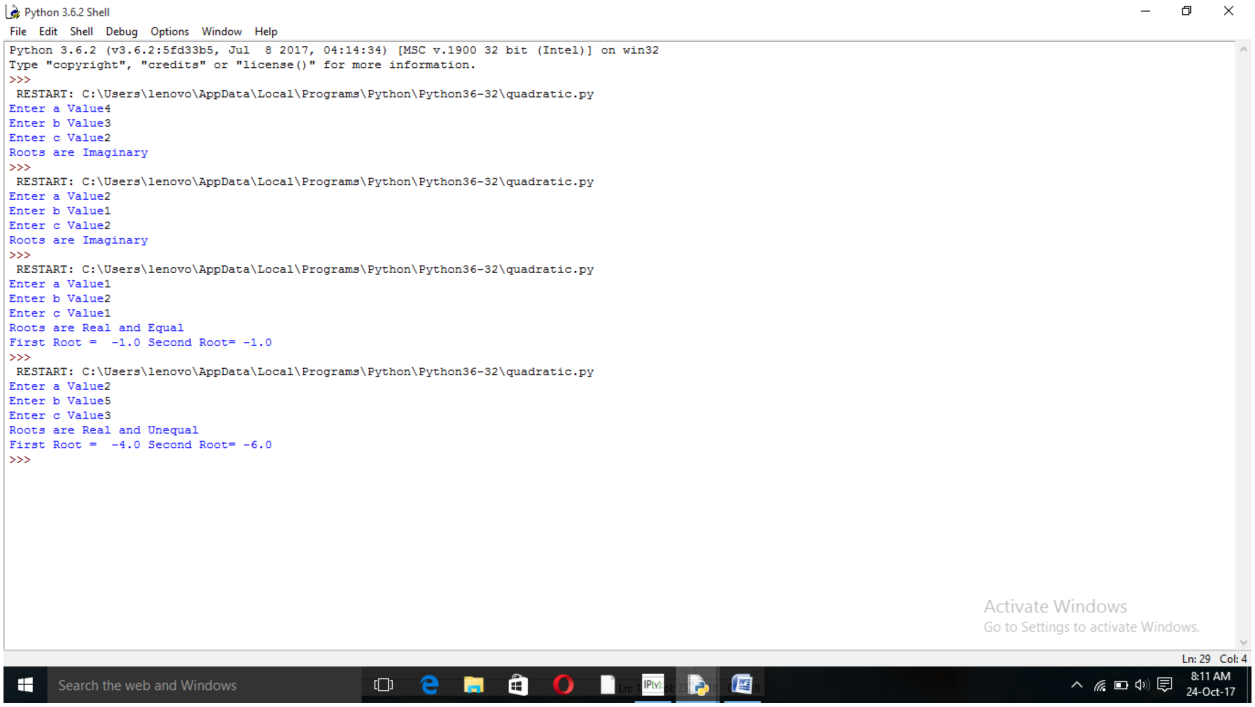


```
36-32\arithmatic-class.py
Enter first number:  67
Enter second number:  323
0.  Exit
1.  Add
2.  Subtraction
3.  Multiplication
4.  Division
Enter choice:  1
Add Result:   390
0.  Exit
1.  Add
2.  Subtraction
3.  Multiplication
4.  Division
Enter choice:  2
Sub Result:   -256
0.  Exit
1.  Add
2.  Subtraction
3.  Multiplication
4.  Division
```

8. Write a Python Program for finding the Roots of a Quadratic Equation

```
import math
class Quadratic:
  def __init__(self,a,b,c):
    self.a=a
    self.b=b
    self.c=c
  def disc(self):
    d=self.b*self.b-4*a*c
    if d==0:
      print("Roots are Real and Equal")
      x1=x2=-b/2*self.a
      print("First Root = ",x1,"Second Root=",x2)
    elif d>0:
      print("Roots are Real and Unequal")
      x1=-self.b+math.sqrt(d)/2*self.a
      x2=-self.b-math.sqrt(d)/2*self.a
      print("First Root = ",round(x1,2),"Second Root=",round(x2,2))
    else:
      print("Roots are Imaginary")
a=int(input("Enter a Value"))
b=int(input("Enter b Value"))
c=int(input("Enter c Value"))
q1=Quadratic(a,b,c)
q1.disc()
```

OUTPUT


Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Activate Windows
Go to Settings to activate Windows.
Ln: 29 Col: 4
Search the web and Windows
8:11 AM
24-Oct-17

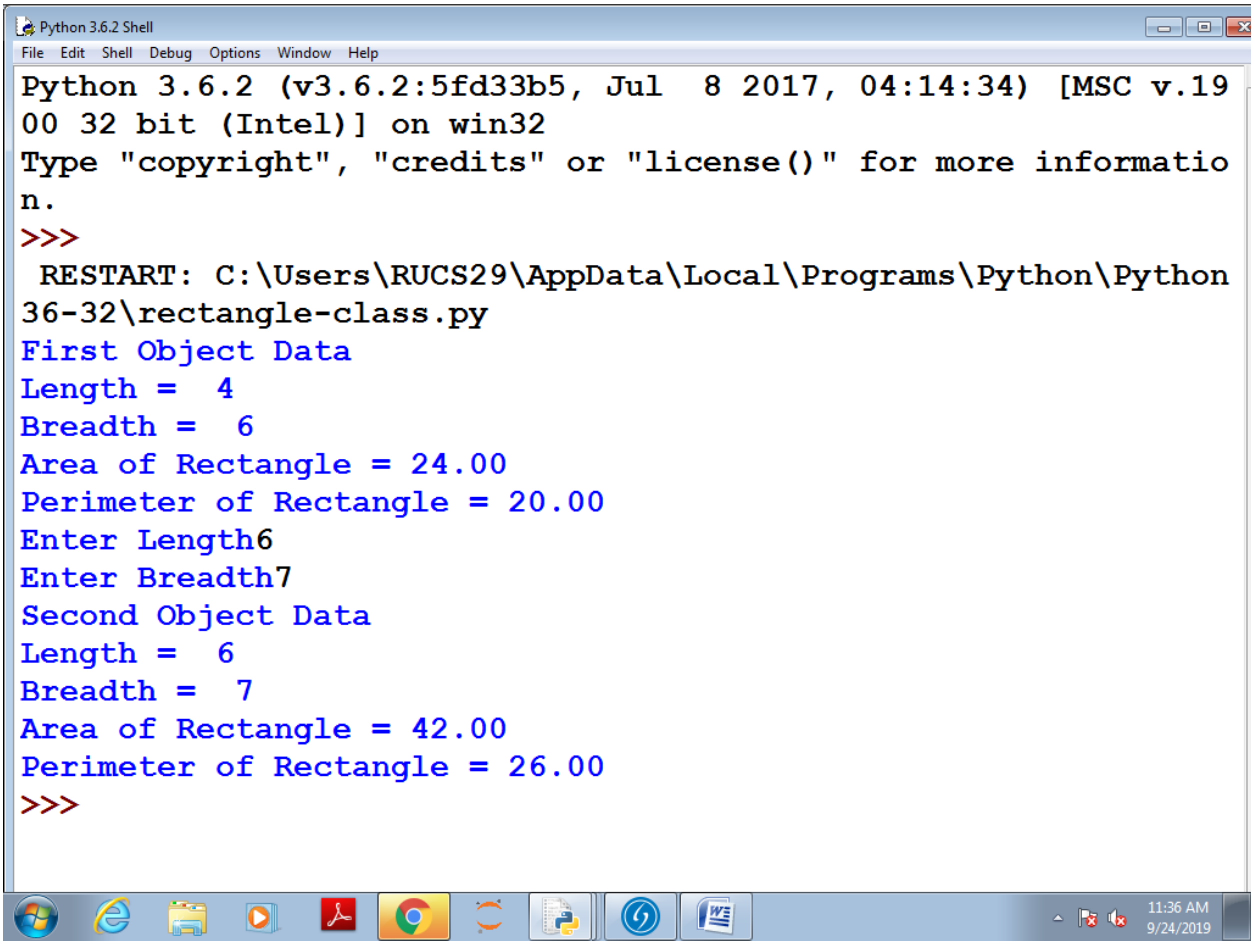
```
Python 3.6.2 (v3.6.2:5fd33b5, Jul  8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
 RESTART: C:\Users\lenovo\AppData\Local\Programs\Python\Python36-32\quadratic.py 
Enter a Value4
Enter b Value3
Enter c Value2
Roots are Imaginary
>>> 
 RESTART: C:\Users\lenovo\AppData\Local\Programs\Python\Python36-32\quadratic.py 
Enter a Value2
Enter b Value1
Enter c Value2
Roots are Imaginary
>>> 
 RESTART: C:\Users\lenovo\AppData\Local\Programs\Python\Python36-32\quadratic.py 
Enter a Value1
Enter b Value2
Enter c Value1
Roots are Real and Equal
First Root =  -1.0 Second Root= -1.0
>>> 
 RESTART: C:\Users\lenovo\AppData\Local\Programs\Python\Python36-32\quadratic.py 
Enter a Value2
Enter b Value5
Enter c Value3
Roots are Real and Unequal
First Root =  -4.0 Second Root= -6.0
>>> 
```

9. Write a Program in Python to find the area and perimeter of a Rectangle using a Class

```
class Rectangle:
   def __init__(self,l,b):
      self.l = l
      self.b = b
   def area(self):
      return self.l * self.b
   def peri(self):
      return 2 *(self.l + self.b)
   def display(self):
      print("Area of Rectangle = %.2f" %(self.area()))
      print("Perimeter of Rectangle = %.2f" %(self.peri()))
r1 = Rectangle(4,6)
print("First Object Data")
print("Length = ",r1.l)
print("Breadth = ",r1.b)
r1.display()
length = eval(input("Enter Length"))
breadth = eval(input("Enter Breadth"))
c2 = Rectangle(length, breadth)
print("Second Object Data")
print("Length = ",c2.l)
print("Breadth = ",c2.b)
```

c2.display()

OUTPUT

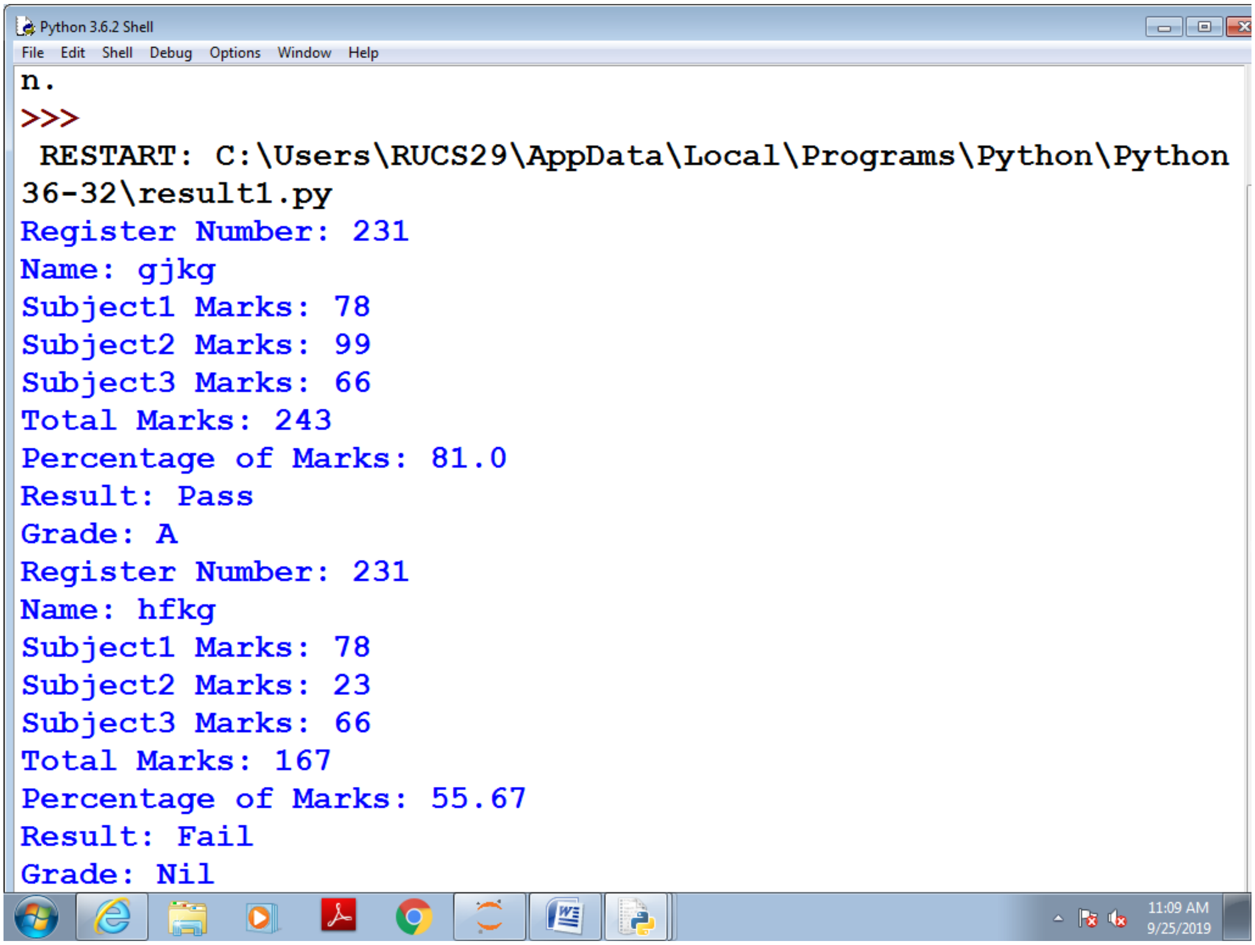


```
Python 3.6.2 (v3.6.2:5fd33b5, Jul  8 2017, 04:14:34) [MSC v.19
00 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more informatio
n.
>>>
 RESTART: C:\Users\RUCS29\AppData\Local\Programs\Python\Python
36-32\rectangle-class.py
First Object Data
Length =  4
Breadth =  6
Area of Rectangle = 24.00
Perimeter of Rectangle = 20.00
Enter Length6
Enter Breadth7
Second Object Data
Length =  6
Breadth =  7
Area of Rectangle = 42.00
Perimeter of Rectangle = 26.00
>>>
```

10. Write a Program in Python to find the Student Result grades using a class

```
class student:
   def __init__(self,rno,name,s1,s2,s3):
      self.rno=rno
      self.name = name
      self.s1=s1
      self.s2=s2
      self.s3=s3
   def tot(self):
      self.t= self.s1 + self.s2 + self.s3
      return self.t
   def percentage(self):
      self.p= self.t/3
      return self.p
   def result(self):
      if self.s1 >=40 and self.s2 >=40 and self.s3 >=40:
         self.r = "Pass"
      else:
         self.r = "Fail"
      return self.r
   def Grade(self):
      if self.r == 'Pass':
          if self.p > 80:
              return 'A'
          elif self.p > 70:
              return 'B'
          elif self.p >60:
              return 'C'
          elif self.p > 50:
              return 'D'
          else:
              return 'E'
      else:
         return 'Nil'
   def output(self):
      print("Register Number:",self.rno)
      print("Name:",self.name)
      print("Subject1 Marks:",self.s1)
      print("Subject2 Marks:",self.s2)
      print("Subject3 Marks:",self.s3)
      print("Total Marks:",self.tot())
      print("Percentage of Marks:",round(self.percentage(),2))
      print("Result:",self.result())
      print("Grade:",self.Grade())
m=student(231,'gjkg',78,99,66)
m.output()
m1=student(231,'hfkg',78,23,66)
m1.output()
```

OUTPUT



```
n.
>>>
 RESTART: C:\Users\RUCS29\AppData\Local\Programs\Python\Python
36-32\result1.py
Register Number: 231
Name: gjkg
Subject1 Marks: 78
Subject2 Marks: 99
Subject3 Marks: 66
Total Marks: 243
Percentage of Marks: 81.0
Result: Pass
Grade: A
Register Number: 231
Name: hfkg
Subject1 Marks: 78
Subject2 Marks: 23
Subject3 Marks: 66
Total Marks: 167
Percentage of Marks: 55.67
Result: Fail
Grade: Nil
```

11. Write a Program in Python to Student Result Processing

```
class student:
  def __init__(self,rno,s1,s2,s3):
    self.s1=s1
    self.s2=s2
    self.s3=s3
  def result(self):
    if self.s1 >=40 and self.s2>=40 and self.s3>=40:
      return 'pass'
    else:
      return 'fail'
  def tot(self):
    self.total=self.s1+self.s2+self.s3
    return self.total
  def average(self):
    return self.total/3
  def disp(self):
    print('sub1 Marks:',self.s1)
    print('sub2 Marks:',self.s2)
    print('sub3 Marks:',self.s3)

st1=student(101,67,87,72)
r=st1.result()
t=st1.tot()
```

print('Marks')

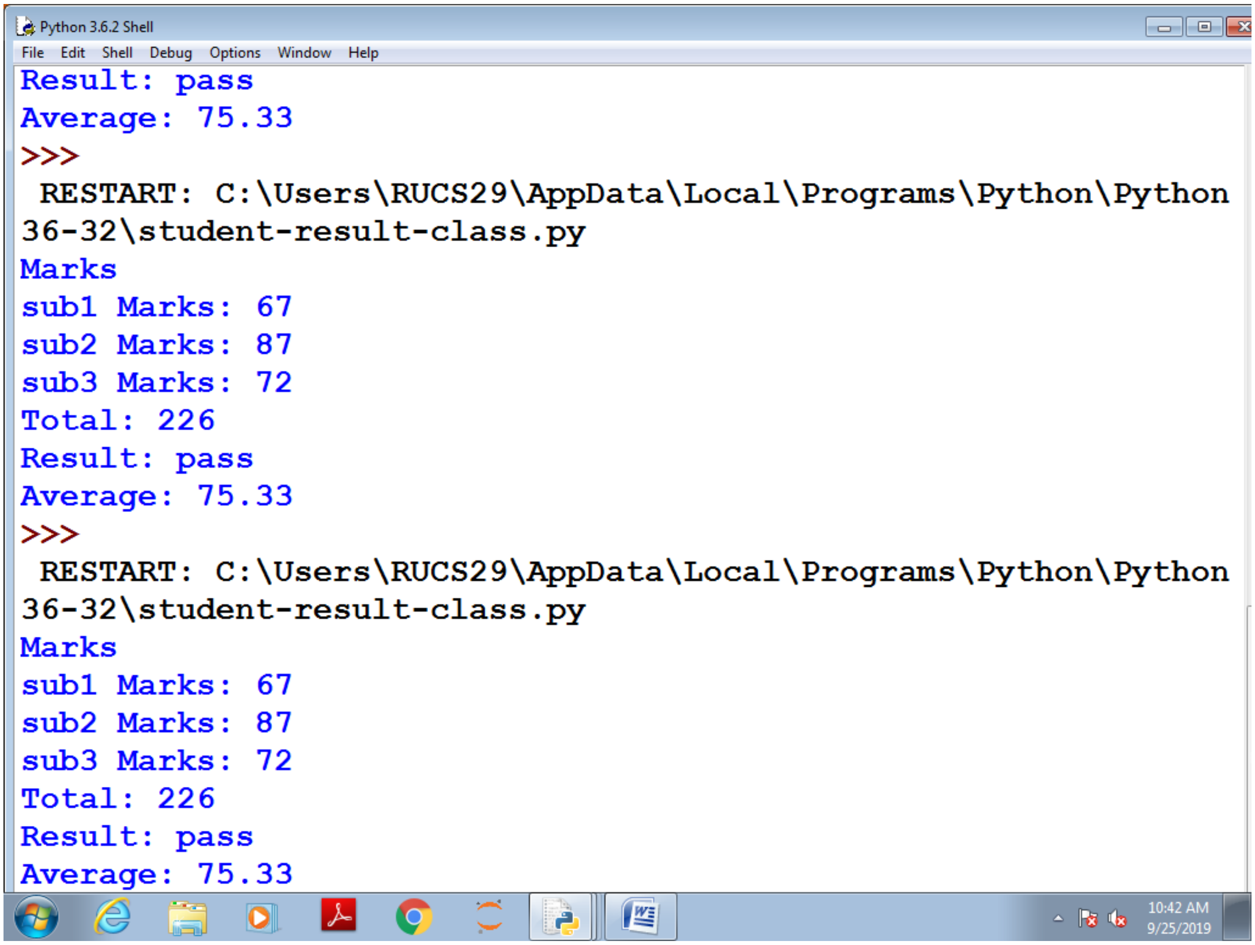
st1.disp()

print('Total:',t)

print('Result:',r)

print('Average:',round(st1.average(),2))

OUTPUT

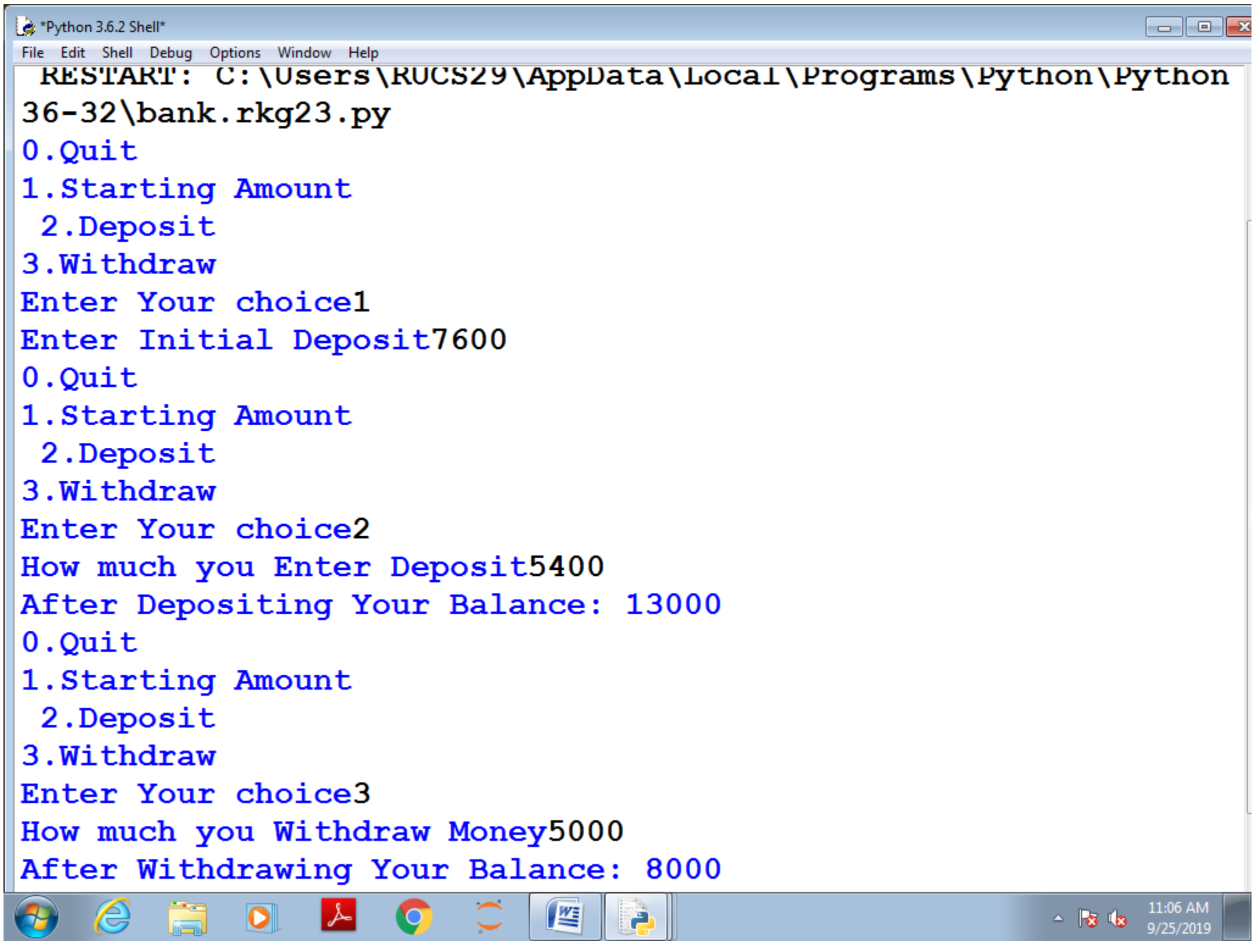


```
Result: pass
Average: 75.33
>>> 
 RESTART: C:\Users\RUCS29\AppData\Local\Programs\Python\Python
36-32\student-result-class.py 
Marks
sub1 Marks: 67
sub2 Marks: 87
sub3 Marks: 72
Total: 226
Result: pass
Average: 75.33
>>> 
 RESTART: C:\Users\RUCS29\AppData\Local\Programs\Python\Python
36-32\student-result-class.py 
Marks
sub1 Marks: 67
sub2 Marks: 87
sub3 Marks: 72
Total: 226
Result: pass
Average: 75.33
```

12. Write a Program in Python to deposit and withdraw transactions of Bank

```
class Bank:
    def __init__(self,bal):
        self.balance = bal
    def Deposit(self,amount):
        self.balance += amount
    def Withdraw(self,amount):
        if self.balance > amount:
             self.balance -= amount
        else:
            print("Insufficient Amount in your Account")
    def get_balance(self):
        return self.balance
choice = 1
while choice !=0:
    print("0.Quit\n1.Starting Amount\n 2.Deposit\n3.Withdraw")
    choice = int(input("Enter Your choice"))
    if choice == 1:
        start_Amount = eval(input("Enter Initial Deposit"))
        bank1 = Bank(start_Amount)
    elif choice == 2:
        Dep_Amount = eval(input("How much you Enter Deposit"))
        bank1.Deposit(Dep_Amount)
        print("After Depositing Your Balance:",bank1.get_balance())
    elif choice == 3:
        With_Amount = eval(input("How much you Withdraw Money"))
        bank1.Withdraw(With_Amount)
        print("After Withdrawing Your Balance:",bank1.get_balance())
    elif choice == 0:
        print("Quiting")
    else:
        print("Invalid Choice")
```

OUTPUT



```
 RESTART: C:\Users\RUCS29\AppData\Local\Programs\Python\Python
36-32\bank.rkg23.py
0.Quit
1.Starting Amount
 2.Deposit
3.Withdraw
Enter Your choice1
Enter Initial Deposit7600
0.Quit
1.Starting Amount
 2.Deposit
3.Withdraw
Enter Your choice2
How much you Enter Deposit5400
After Depositing Your Balance: 13000
0.Quit
1.Starting Amount
 2.Deposit
3.Withdraw
Enter Your choice3
How much you Withdraw Money5000
After Withdrawing Your Balance: 8000
```

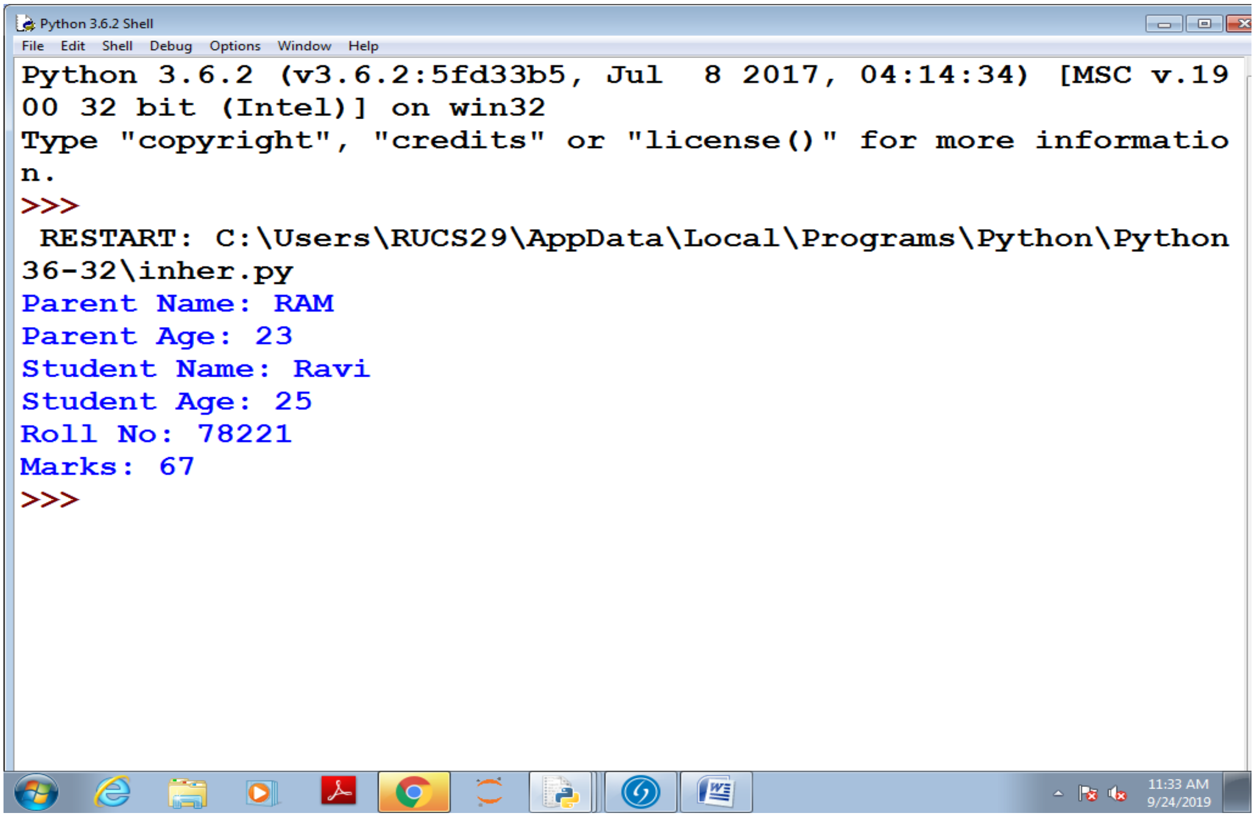
13.Write a Program in Python to implement a Single Inheritance

```
class person(object):
    def __init__(self,name,age):
         self.name=name
         self.age=age
    def getName(self):
        return self.name
    def getAge(self):
         return self.age
class student(person):
      def __init__(self,name,age,rollno,Marks):
             person.__init__ (self, name, age)
             self.rollno=rollno
             self.Marks=Marks
      def getRoll(self):
           return self.rollno
      def getMarks(self):
           return self.Marks
p1 = person("RAM",23)
print("Parent Name:",p1.getName())
print("Parent Age:",p1.getAge())
s1 = student("Ravi",25,78221,67)
print("Student Name:",s1.getName())
print("Student Age:",s1.getAge())
```

print("Roll No:",s1.getRoll())

print("Marks:",s1.getMarks())

OUTPUT



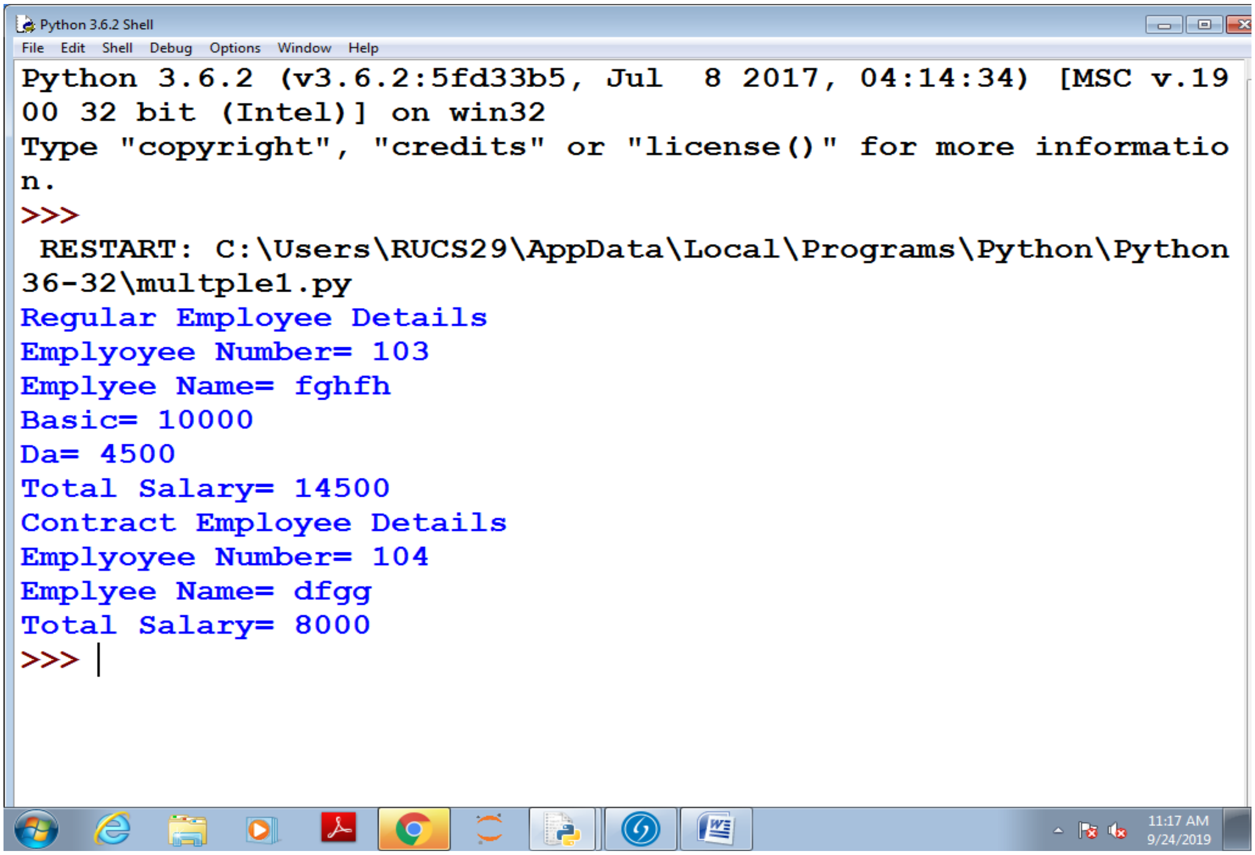
```
Python 3.6.2 (v3.6.2:5fd33b5, Jul  8 2017, 04:14:34) [MSC v.19
00 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more informatio
n.
>>> 
 RESTART: C:\Users\RUCS29\AppData\Local\Programs\Python\Python
36-32\inher.py
Parent Name: RAM
Parent Age: 23
Student Name: Ravi
Student Age: 25
Roll No: 78221
Marks: 67
>>> 
```

14. Write a Program in Python to display employee details using Multiple Inheritance

```
class employee:
   def __init__(self,Empid,name):
      self.Empid = Empid
      self.name = name
   def getEmpid(self):
      return self.Empid
   def getName(self):
      return self.name
class regular(employee):
   def __init__(self,Empid,name,basic,da):
      employee.__init__(self,Empid,name)
      self.basic = basic
      self.da=da
   def total_salary(self):
      return self.basic + self.da
class contract(employee):
   def __init__(self,Empid,name,cons):
      employee.__init__(self,Empid,name)
      self.cons = cons
   def gross_salary(self):
      return self.cons
r  = regular(103,"fghfh",10000,4500)
```

```
c = contract(104,"dfgg",8000)

print("Regular Employee Details")

print("Emplyoyee Number=",r.getEmpid())

print("Emplyee Name=",r.getName())

print("Basic=",r.basic)

print("Da=",r.da)

print("Total Salary=",r.total_salary())

print("Contract Employee Details")

print("Emplyoyee Number=",c.getEmpid())

print("Emplyee Name=",c.getName())

print("Total Salary=",c.gross_salary())
```

OUTPUT



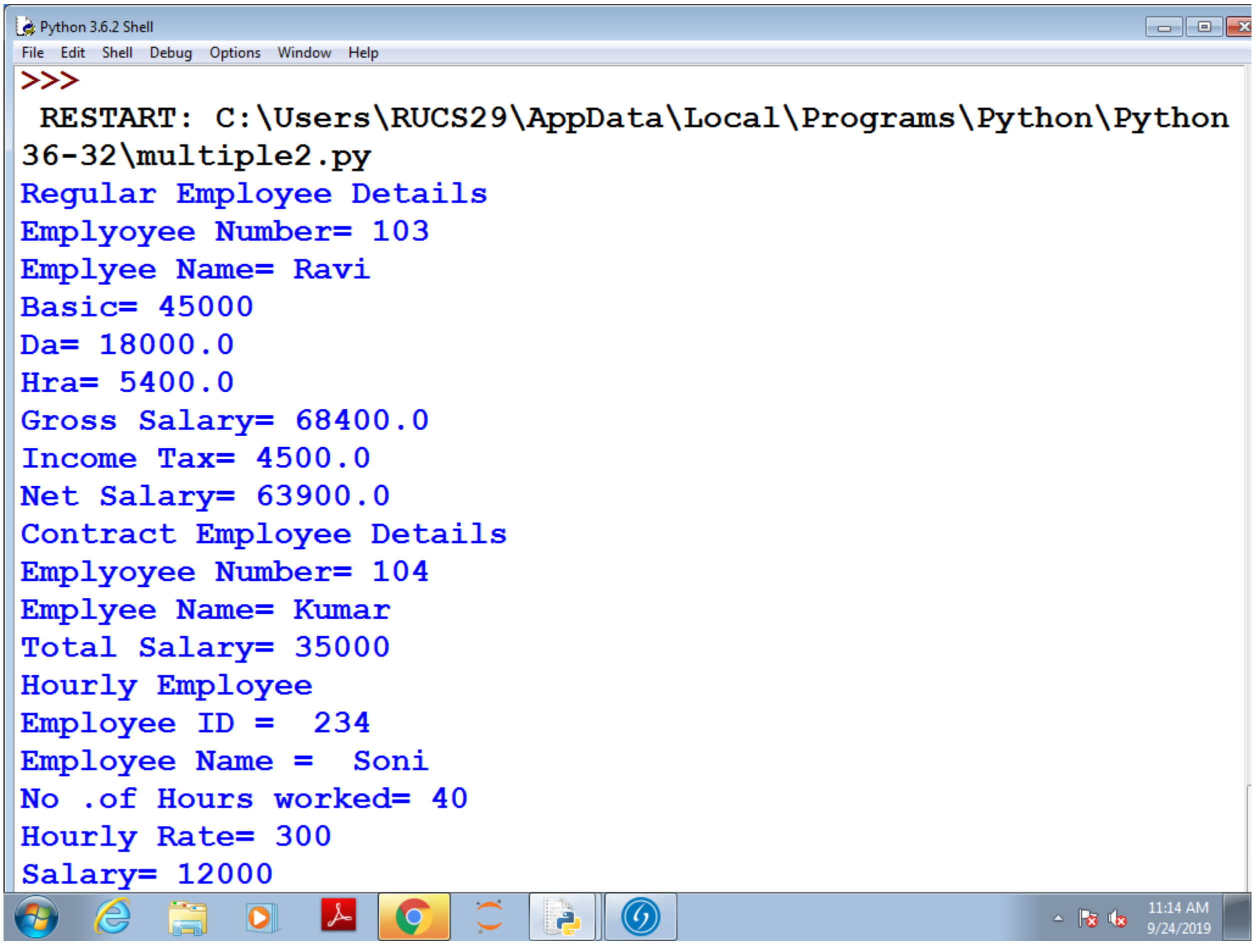
```
Python 3.6.2 (v3.6.2:5fd33b5, Jul  8 2017, 04:14:34) [MSC v.19
00 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more informatio
n.
>>> 
 RESTART: C:\Users\RUCS29\AppData\Local\Programs\Python\Python
36-32\multple1.py
Regular Employee Details
Emplyoyee Number= 103
Emplyee Name= fghfh
Basic= 10000
Da= 4500
Total Salary= 14500
Contract Employee Details
Emplyoyee Number= 104
Emplyee Name= dfgg
Total Salary= 8000
>>> |
```

15.Write a Program in Python to display employee salary details using Multiple Inheritance

```
class employee:
    def __init__(self,Empid,name):
        self.Empid = Empid
        self.name = name
    def getEmpid(self):
        return self.Empid
    def getName(self):
        return self.name
class regular(employee):
    def __init__(self,Empid,name,basic):
        employee.__init__(self,Empid,name)
        self.basic = basic
    def get_basic(self):
        return self.basic
    def get_da(self):
        self.da = self.basic * 40/100
        return self.da
    def get_hra(self):
        self.hra = self.basic * 12/100
        return self.hra
    def get_gross(self):
        self.gross = self.basic + self.da + self.hra
        return self.gross
    def get_itax(self):
        self.itax = self.basic * 10/100
        return self.itax
    def get_net(self):
        self.netsal = self.gross - self.itax
        return self.netsal
class contract(employee):
    def __init__(self,Empid,name,consolidate):
        employee.__init__(self,Empid,name)
        self.consolidate = consolidate
    def gross_salary(self):
        return self.consolidate
class Hourly_emp(employee):
    def __init__(self,Empid,name,No_Hours,hrate):
        employee.__init__(self,Empid,name)
        self.No_Hours = No_Hours
        self.hrate = hrate
    def get_Hours(self):
        return self.No_Hours
    def get_Hrate(self):
        return self.hrate
    def Hsal(self):
        self.hsal = self.No_Hours * self.hrate
        return self.hsal
```

```
r  = regular(103,"Ravi",45000)

c = contract(104,"Kumar",35000)

h = Hourly_emp(234,"Soni",40,300)

print("Regular Employee Details")

print("Emplyoyee Number=",r.getEmpid())

print("Emplyee Name=",r.getName())

print("Basic=",r.basic)

print("Da=",r.get_da())

print("Hra=",r.get_hra())

print("Gross Salary=",r.get_gross())

print("Income Tax=",r.get_itax())

print("Net Salary=",r.get_net())

print("Contract Employee Details")

print("Emplyoyee Number=",c.getEmpid())

print("Emplyee Name=",c.getName())

print("Total Salary=",c.gross_salary())

print("Hourly Employee")

print("Employee ID = ",h.getEmpid())

print("Employee Name = ",h.getName())

print("No .of Hours worked=",h.get_Hours())

print("Hourly Rate=",h.get_Hrate())

print("Salary=",h.Hsal())
```

OUTPUT

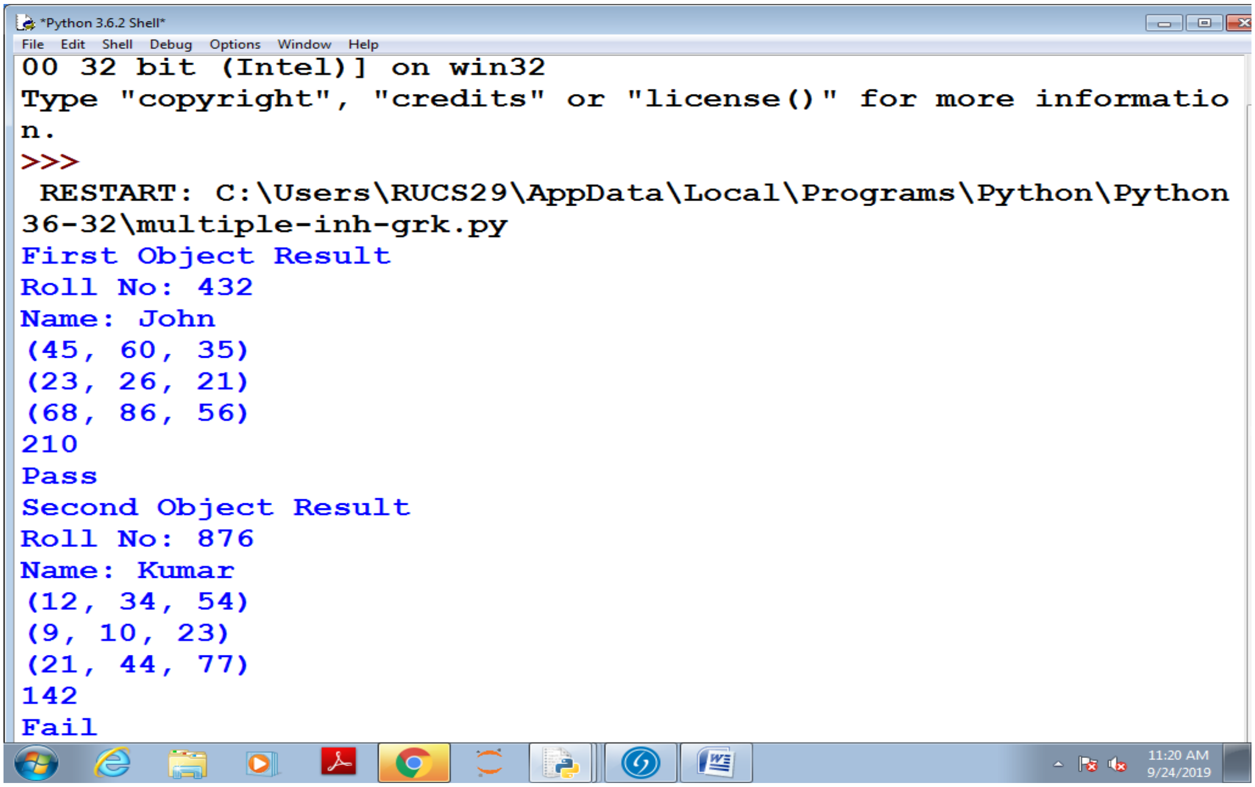


```
>>>
 RESTART: C:\Users\RUCS29\AppData\Local\Programs\Python\Python
36-32\multiple2.py
Regular Employee Details
Emplyoyee Number= 103
Emplyee Name= Ravi
Basic= 45000
Da= 18000.0
Hra= 5400.0
Gross Salary= 68400.0
Income Tax= 4500.0
Net Salary= 63900.0
Contract Employee Details
Emplyoyee Number= 104
Emplyee Name= Kumar
Total Salary= 35000
Hourly Employee
Employee ID =  234
Employee Name =  Soni
No .of Hours worked= 40
Hourly Rate= 300
Salary= 12000
```

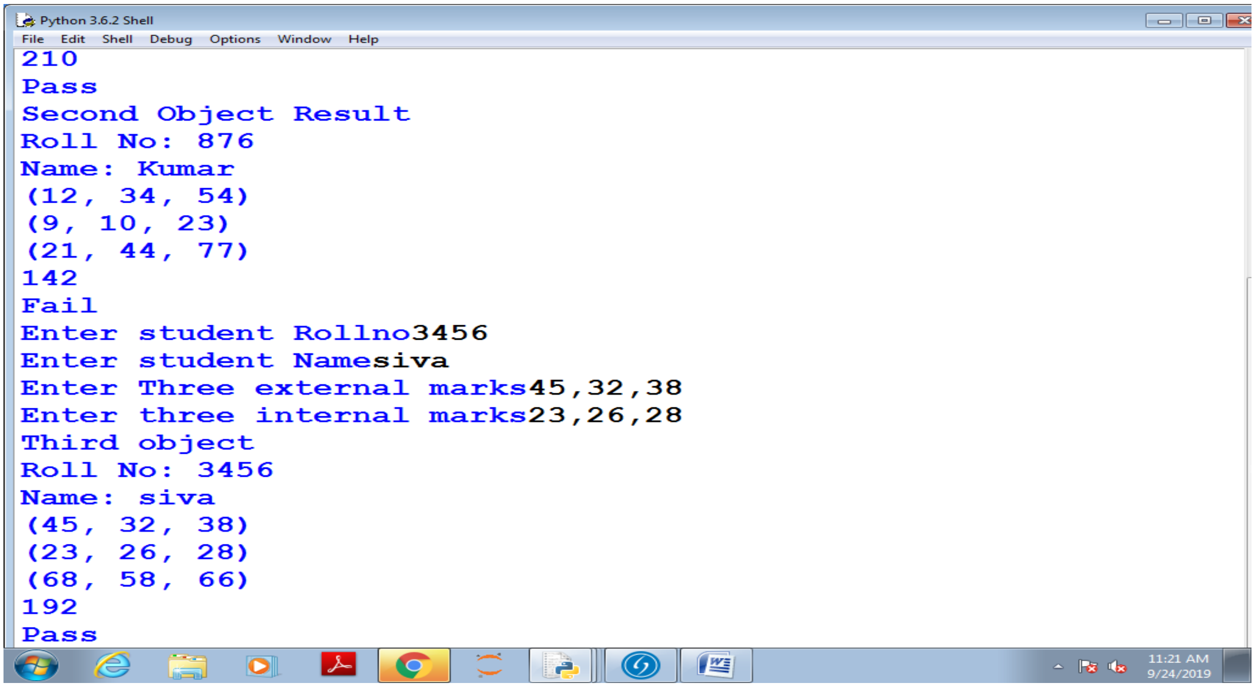
16. Write a Program in Python to perform student result processing using a Hybrid Inheritance

```
class student:
   def __init__(self, rno,name):
      self.name = name
      self.rno = rno
   def display1 (self):
      print("Roll No:",self.rno)
      print("Name:",self.name)
class Ext_marks(student):
   def __init__(self, es1,es2,es3):
      self.es1 = es1
      self.es2 = es2
      self.es3 = es3
   def getExt_marks(self):
      return self.es1, self.es2, self.es3
class Int_marks(student):
   def __init__(self, is1,is2,is3):
      self.is1 = is1
      self.is2 = is2
      self.is3= is3
   def getInt_marks(self):
      return self.is1, self.is2, self.is3
class Result(Ext_marks, Int_marks):
    def __init__(self, name, rno, es1,es2,es3,is1,is2,is3):
        student.__init__(self, name, rno)
        Ext_marks.__init__(self, es1,es2,es3)
        Int_marks.__init__(self, is1,is2,is3)
    def tot(self):
         self.ts1 = self.es1 + self.is1
         self.ts2 = self.es2 + self.is2
         self.ts3 = self.es3 + self.is3
         return self.ts1, self.ts2, self.ts3
    def tot1(self):
         self.totmarks = self.ts1 + self.ts2 + self.ts3
         return self.totmarks
    def res(self):
        if self.ts1 > 40 and self.ts2 > 40 and self.ts3>40:
            return "Pass"
        else:
            return "Fail"
t1 = Result(432,'John',45,60,35,23,26,21)
t2 = Result(876,"Kumar",12,34,54,9,10,23)
print("First Object Result")
t1.display1()
print(t1.getExt_marks())
print(t1.getInt_marks())
print(t1.tot())
print(t1.tot1())
print(t1.res())
print("Second Object Result")
t2.display1()
print(t2.getExt_marks())
print(t2.getInt_marks())
print(t2.tot())
print(t2.tot1())
print(t2.res())
RollNo=eval(input("Enter student Rollno"))
Name=input("Enter student Name")
s1,s2,s3=eval(input("Enter Three external marks"))
i1,i2,i3=eval(input("Enter three internal marks"))
t3=Result(RollNo,Name,s1,s2,s3,i1,i2,i3)
print("Third object")
t3.display1()
print(t3.getExt_marks())
print(t3.getInt_marks())
print(t3.tot())
print(t3.tot1())
print(t3.res())
```

OUTPUT



```
00 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more informatio
n.
>>>
 RESTART: C:\Users\RUCS29\AppData\Local\Programs\Python\Python
36-32\multiple-inh-grk.py
First Object Result
Roll No: 432
Name: John
(45, 60, 35)
(23, 26, 21)
(68, 86, 56)
210
Pass
Second Object Result
Roll No: 876
Name: Kumar
(12, 34, 54)
(9, 10, 23)
(21, 44, 77)
142
Fail
```



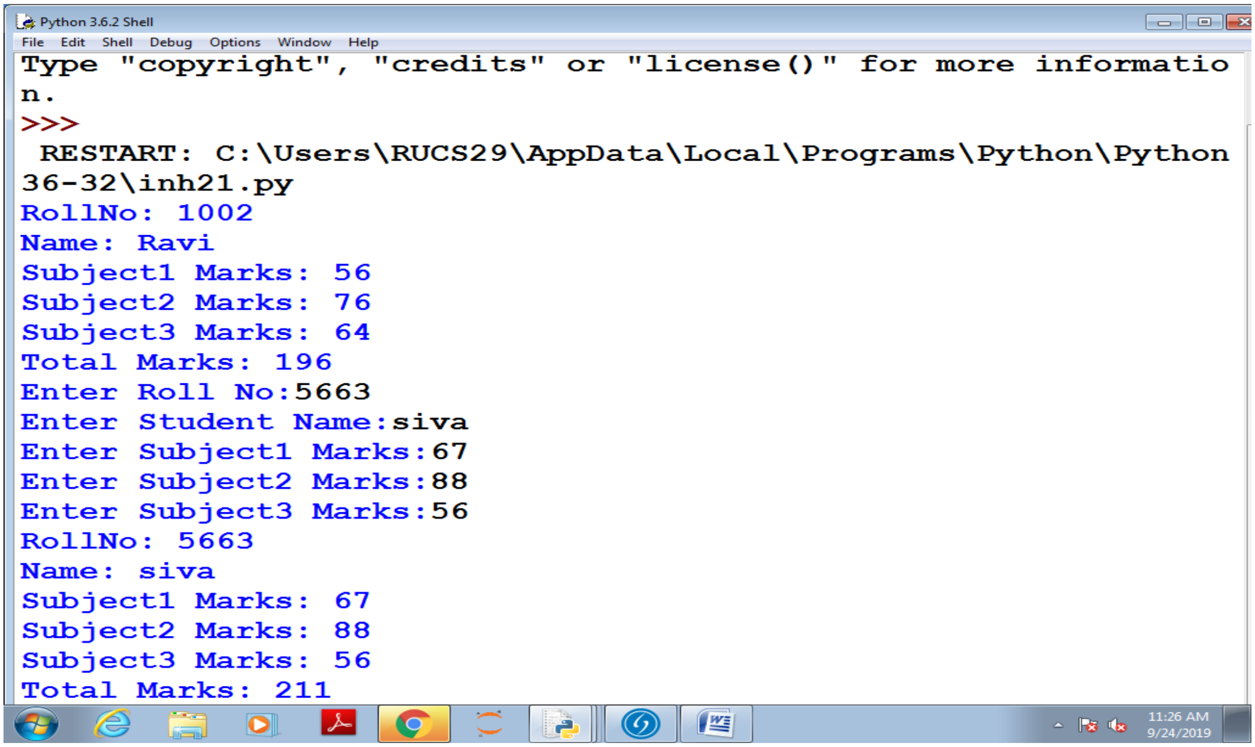
```
210
Pass
Second Object Result
Roll No: 876
Name: Kumar
(12, 34, 54)
(9, 10, 23)
(21, 44, 77)
142
Fail
Enter student Rollno3456
Enter student Namesiva
Enter Three external marks45,32,38
Enter three internal marks23,26,28
Third object
Roll No: 3456
Name: siva
(45, 32, 38)
(23, 26, 28)
(68, 58, 66)
192
Pass
```

17. Write a Program in Python to Display Student results using a Multi level Inheritance

```
class student:
   def __init__(self,rno,name):
      self.rno=rno
      self.name=name
   def display(self):
      print("RollNo:",self.rno)
      print("Name:",self.name)
class marks(student):
   def __init__(self,rno,name,s1,s2,s3):
      student.__init__(self,rno,name)
      self.s1=s1
      self.s2=s2
      self.s3=s3
   def disp_marks(self):
      print("Subject1 Marks:",self.s1)
      print("Subject2 Marks:",self.s2)
      print("Subject3 Marks:",self.s3)
class total(marks):
   def sum(self):
      return self.s1+self.s2+self.s3
t=total(1002,"Ravi",56,76,64)
t.display()
```

```
t.disp_marks()

t.sum()

print("Total Marks:",t.sum())

rno = eval(input("Enter Roll No:"))

name = input("Enter Student Name:")

s1 = eval(input("Enter Subject1 Marks:"))

s2 = eval(input("Enter Subject2 Marks:"))

s3 = eval(input("Enter Subject3 Marks:"))

t2 = total(rno,name,s1,s2,s3)

t2.display()

t2.disp_marks()

print("Total Marks:",t2.sum())
```

OUTPUT

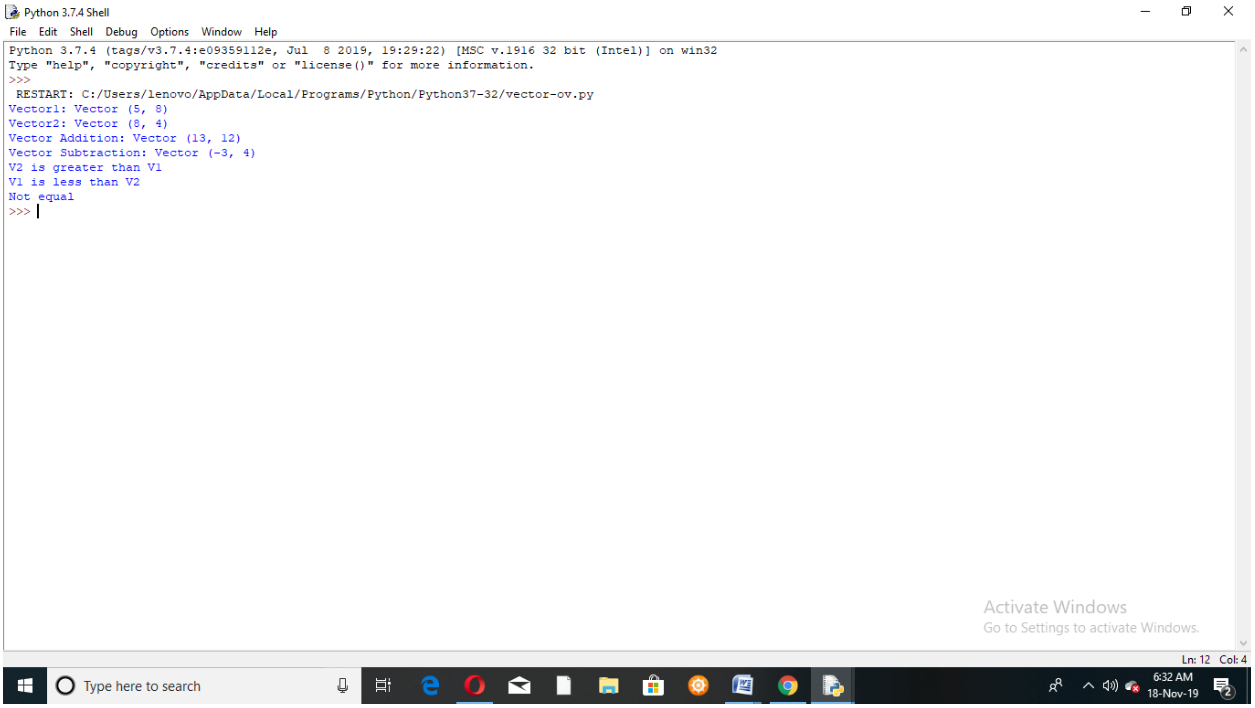


```
Type "copyright", "credits" or "license()" for more informatio
n.
>>>
 RESTART: C:\Users\RUCS29\AppData\Local\Programs\Python\Python
36-32\inh21.py
RollNo:  1002
Name:  Ravi
Subject1 Marks:  56
Subject2 Marks:  76
Subject3 Marks:  64
Total Marks:  196
Enter Roll No:5663
Enter Student Name:siva
Enter Subject1 Marks:67
Enter Subject2 Marks:88
Enter Subject3 Marks:56
RollNo:  5663
Name:  siva
Subject1 Marks:  67
Subject2 Marks:  88
Subject3 Marks:  56
Total Marks:  211
```

18.Write a Program in Python to perform Vector operations using Operator overloading

```
class Vector:
    def __init__(self, a, b):
        self.a = a
        self.b = b
    def __str__(self):
        return 'Vector (%d, %d)' % (self.a, self.b)
    def __add__(self,other):
        return Vector(self.a + other.a, self.b + other.b)
    def __sub__(self, other):
        return Vector(self.a - other.a, self.b - other.b)
    def __gt__(self, other):
        if(self.a > other.a):
            return True
        else:
            return False
    def __lt__(self, other):
        if(self.a < other.a):
            return True
        else:
            return False
    def __eq__(self, other):
        if(self.a == other.a):
            return "Both are equal"
        else:
            return "Not equal"
v1 = Vector(5,8)
v2 = Vector(8,4)
print("Vector1:",v1)
print("Vector2:",v2)
print ("Vector Addition:", v1 + v2)
print ("Vector Subtraction:",v1 - v2)
if(v1 > v2):
    print("V1 is greater than V2")
else:
    print("V2 is greater than V1")
if(v1 < v2):
    print("V1 is less than V2")
else:
    print("V2 is less than V1")
print(v1 == v2)
```

OUTPUT



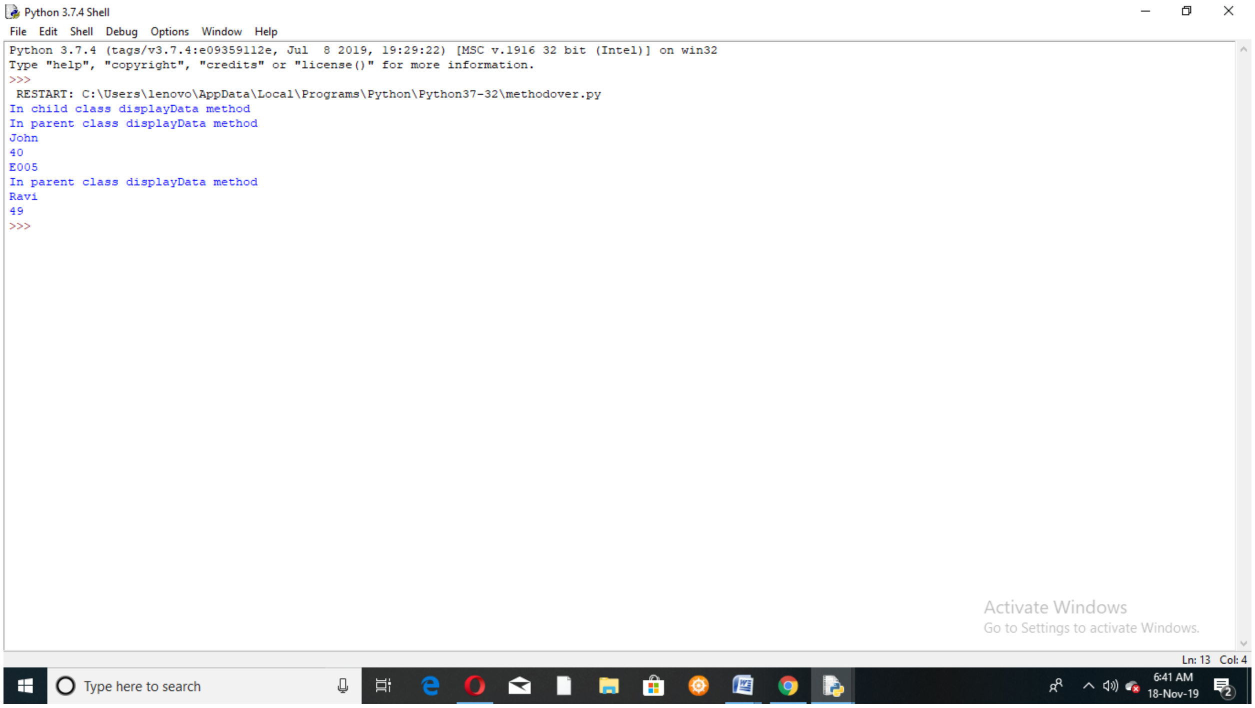
```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
 RESTART: C:/Users/lenovo/AppData/Local/Programs/Python/Python37-32/vector-ov.py
Vector1: Vector (5, 8)
Vector2: Vector (8, 4)
Vector Addition: Vector (13, 12)
Vector Subtraction: Vector (-3, 4)
V2 is greater than V1
V1 is less than V2
Not equal
>>> 
```

19.Write a Program in Python to implement Method overriding

```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

  def displayData(self):
    print('In parent class displayData method')
    print(self.name)
    print(self.age)

class Employee(Person):
  def __init__(self, name, age, id):
    # calling constructor of super class
    super().__init__(name, age)
    self.empId = id

  def displayData(self):
    print('In child class displayData method')
    Person.displayData(self)
    print(self.empId)
#Employee class object
emp1 = Person("Ravi",49)
emp = Employee('John', 40, 'E005')
emp.displayData()
```

emp1.displayData()

OUTPUT

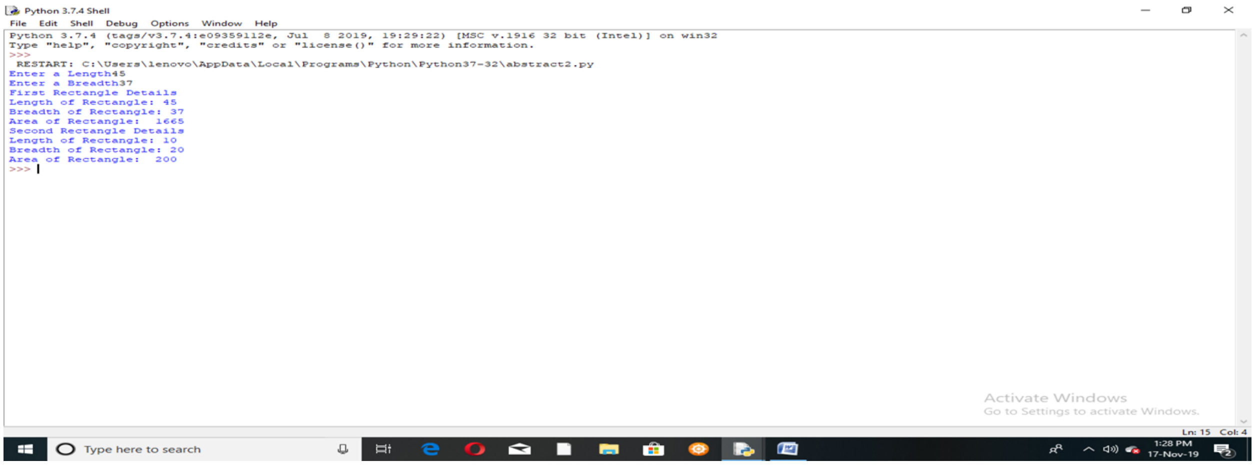


```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
 RESTART: C:\Users\lenovo\AppData\Local\Programs\Python\Python37-32\methodover.py
In child class displayData method
In parent class displayData method
John
40
E005
In parent class displayData method
Ravi
49
>>> 
```

20. Write a Program in Python to find the area of Rectangle with a given length and breadth using Abstract Class

```
from abc import ABC, abstractmethod
class Shape(ABC):
  @abstractmethod
  def area(self):
    pass
class Rectangle(Shape):
  def __init__(self, x,y):
    self.l = x
    self.b=y
  def area(self):
    return self.l*self.b
  def disp(self):
      print ('Area of Rectangle: ',self.area())
l = eval(input("Enter a Length"))
b = eval(input("Enter a Breadth"))
r1 = Rectangle(l,b)
print("First Rectangle Details")
print("Length of Rectangle:",l)
print("Breadth of Rectangle:",b)
r1.disp()
r2 = Rectangle(10,20)
print("Second Rectangle Details")
print("Length of Rectangle:",r2.l)
print("Breadth of Rectangle:",r2.b)
r2.disp()
```

OUTPUT

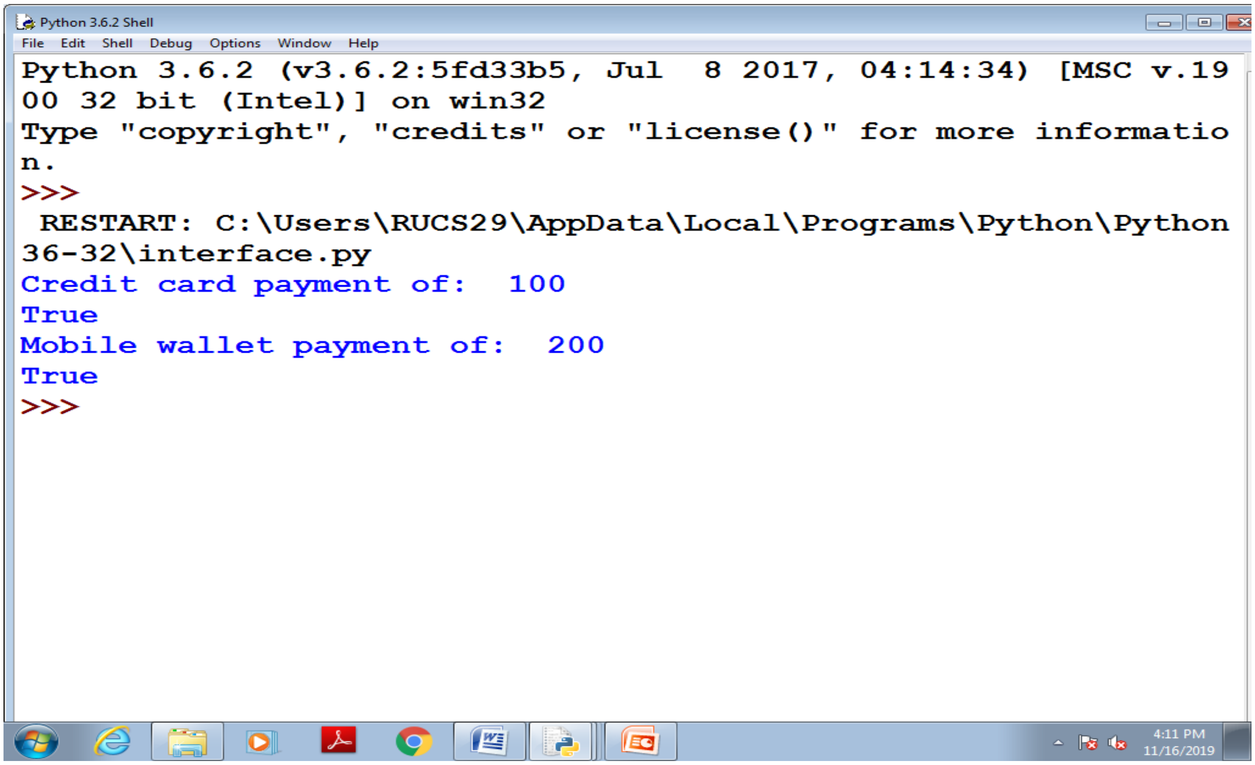


```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
 RESTART: C:\Users\lenovo\AppData\Local\Programs\Python\Python37-32\abstract2.py
Enter a Length45
Enter a Breadth37
First Rectangle Details
Length of Rectangle: 45
Breadth of Rectangle: 37
Area of Rectangle:  1665
Second Rectangle Details
Length of Rectangle: 10
Breadth of Rectangle: 20
Area of Rectangle:  200
>>> 
```

21. Write a Program in Python to display an amount of credit card or Mobile wallet using an Interface

```
from abc import ABC, abstractmethod

class Payment(ABC):

   @abstractmethod

   def payment(self, amount):

      pass


class CreditCardPayment(Payment):

   def payment(self, amount):

      print('Credit card payment of: ', amount)

class MobileWalletPayment(Payment):

   def payment(self, amount):

      print('Mobile wallet payment of: ', amount)


obj = CreditCardPayment()

obj.payment(100)

print(isinstance(obj, Payment))

obj = MobileWalletPayment()

obj.payment(200)

print(isinstance(obj, Payment))
```

OUTPUT

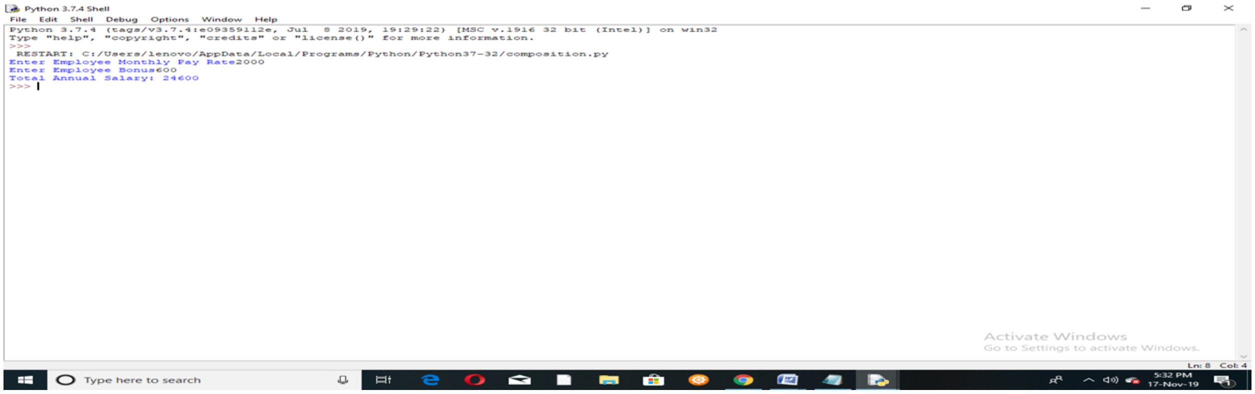


```
Python 3.6.2 (v3.6.2:5fd33b5, Jul  8 2017, 04:14:34) [MSC v.19
00 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more informatio
n.
>>>
 RESTART: C:\Users\RUCS29\AppData\Local\Programs\Python\Python
36-32\interface.py
Credit card payment of:  100
True
Mobile wallet payment of:  200
True
>>>
```

22.Write a Python Program to demonstrate a Composition in OOPS

```
class Salary:
    def __init__(self, pay):
        self.pay = pay
    def get_total(self):
        return (self.pay*12)
class Employee:
    def __init__(self, pay, bonus):
        self.pay = pay
        self.bonus = bonus
        self.obj_salary = Salary(self.pay)
    def annual_salary(self):
        return "Total Annual Salary: " + str(self.obj_salary.get_total() + self.bonus)
pay = eval(input("Enter Employee Monthly Pay Rate"))
bonus = eval(input("Enter Employee Bonus"))
obj_emp = Employee(pay,bonus)
print(obj_emp.annual_salary())
```

OUTPUT

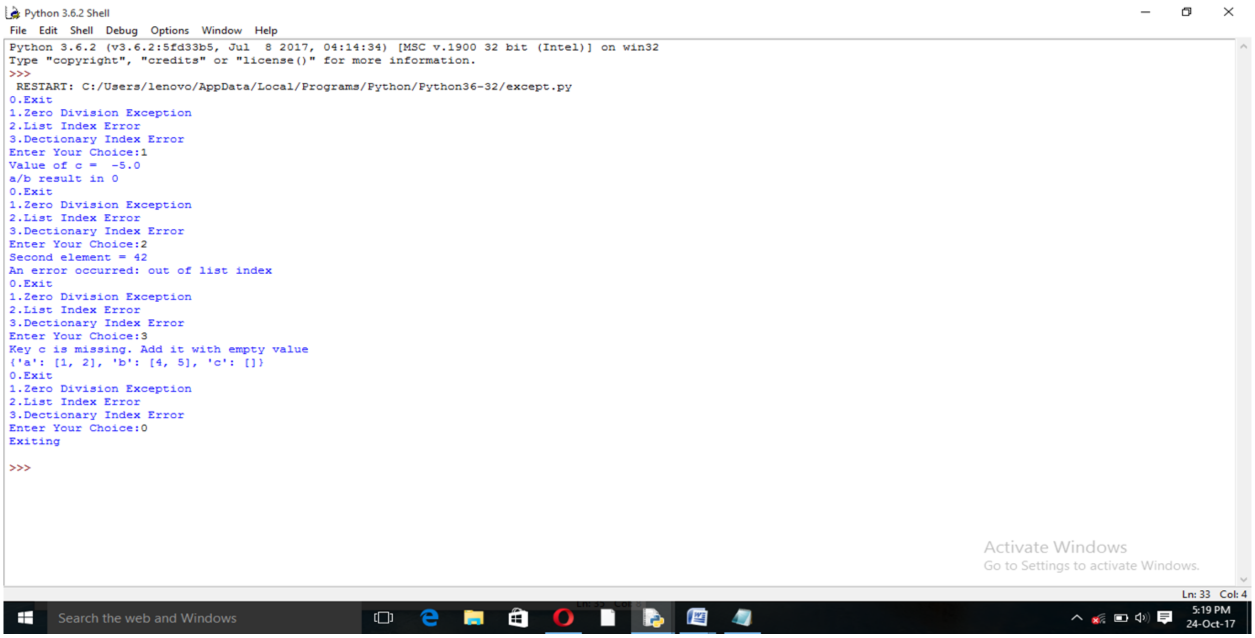


```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
 RESTART: C:/Users/lenovo/AppData/Local/Programs/Python/Python37-32/composition.py
Enter Employee Monthly Pay Rate2000
Enter Employee Bonus600
Total Annual Salary: 24600
>>> 
```

23.Write a Python Program to demonstrate different exception handling (Zero Division, List and Dictionary Index Exceptions)

```
def sample(a , b):
    try:
        c = ((a+b) / (a-b))
    except ZeroDivisionError:
        print ("a/b result in 0")
    else:
        print("Value of c = ",c)
choice=1
while choice !=0:
    print("0.Exit")
    print("1.Zero Division Exception")
    print("2.List Index Error")
    print("3.Dectionary Index Error")
    choice=int(input("Enter Your Choice:"))
    if choice == 1:
        sample(2.0, 3.0)
        sample(3.0, 3.0)
    elif choice ==2:
        a = [23, 42, 38,54]
        try:
            print( "Second element = %d" %(a[1]))
            print ("Fifth element = %d" %(a[4]))
        except IndexError:
          print("An error occurred: out of list index")
    elif choice ==3:
        dct = dict(a=[1, 2], b=[4, 5])
        key = 'c'
        try:
          dct[key]
        except:
          print("Key %s is missing. Add it with empty value" % key)
          dct['c'] = []
        print(dct)
    else:
        print("Exiting")
print()
```

Output



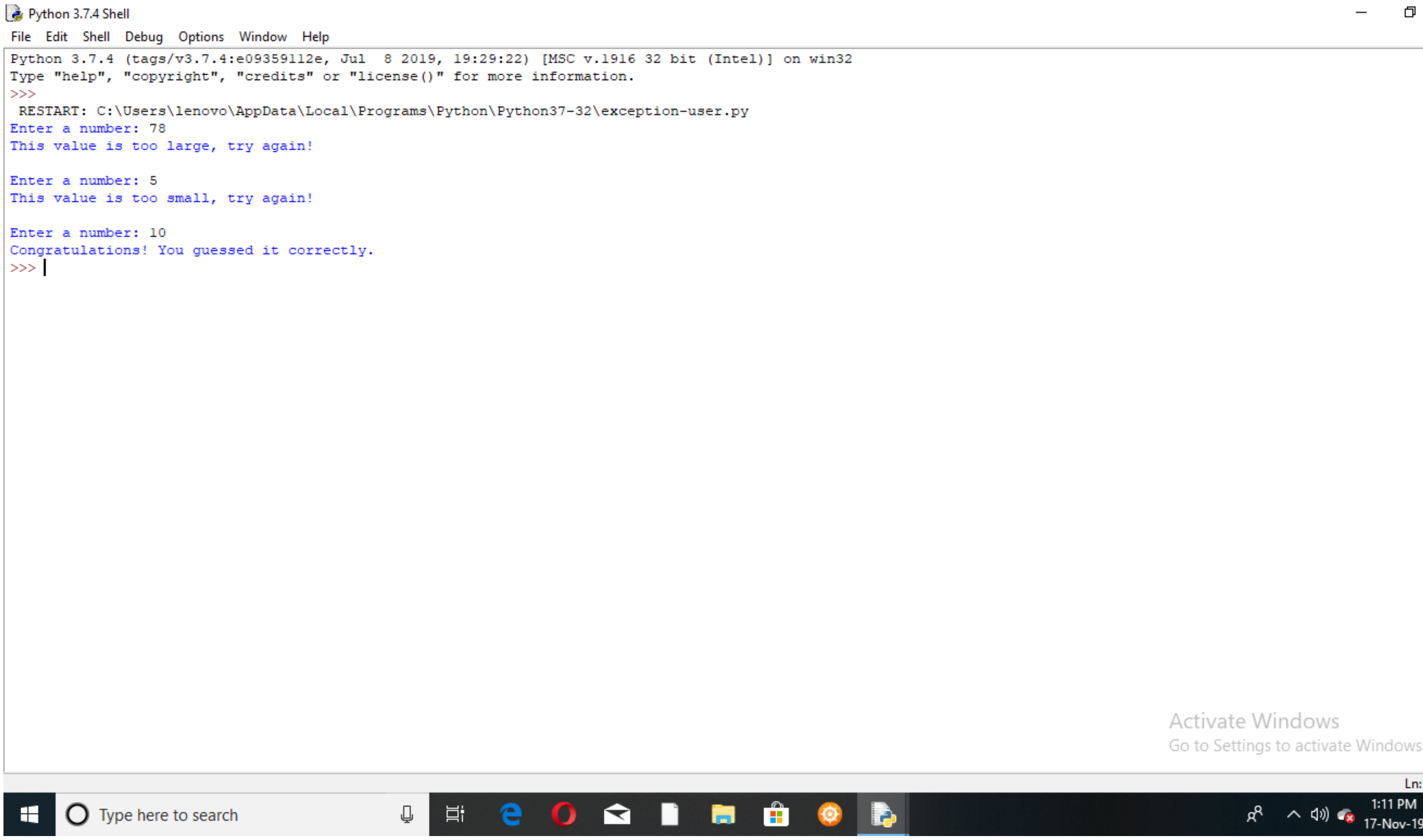
```
Python 3.6.2 (v3.6.2:5fd33b5, Jul  8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
 RESTART: C:/Users/lenovo/AppData/Local/Programs/Python/Python36-32/except.py
0.Exit
1.Zero Division Exception
2.List Index Error
3.Dectionary Index Error
Enter Your Choice:1
Value of c =  -5.0
a/b result in 0
0.Exit
1.Zero Division Exception
2.List Index Error
3.Dectionary Index Error
Enter Your Choice:2
Second element = 42
An error occurred: out of list index
0.Exit
1.Zero Division Exception
2.List Index Error
3.Dectionary Index Error
Enter Your Choice:3
Key c is missing. Add it with empty value
{'a': [1, 2], 'b': [4, 5], 'c': []}
0.Exit
1.Zero Division Exception
2.List Index Error
3.Dectionary Index Error
Enter Your Choice:0
Exiting

>>> 
```

24.Write a Program in Python to implement user defined Exception

```
class Error(Exception):
    """Base class for other exceptions"""
    pass
class ValueTooSmallError(Error):
    """Raised when the input value is too small"""
    pass
class ValueTooLargeError(Error):
    """Raised when the input value is too large"""
    pass
number = 10
while True:
    try:
        num = int(input("Enter a number: "))
        if num < number:
            raise ValueTooSmallError
        elif num > number:
            raise ValueTooLargeError
        break
    except ValueTooSmallError:
        print("This value is too small, try again!")
        print()
    except ValueTooLargeError:
        print("This value is too large, try again!")
```

```
      print()
print("Congratulations! You guessed it correctly.")
```

OUTPUT



```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
 RESTART: C:\Users\lenovo\AppData\Local\Programs\Python\Python37-32\exception-user.py
Enter a number: 78
This value is too large, try again!

Enter a number: 5
This value is too small, try again!

Enter a number: 10
Congratulations! You guessed it correctly.
>>> 
```