

**S.Y.BSc.IT**

**SEMESTER-III**

**PYTHON PROGRAMMING**

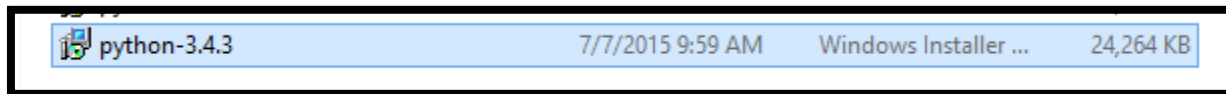
**PRACTICAL**

**MANUAL**

**2017-2018**

## Installation Step for Python 3.4.x and MySQL Connectivity to Python 3.4.x

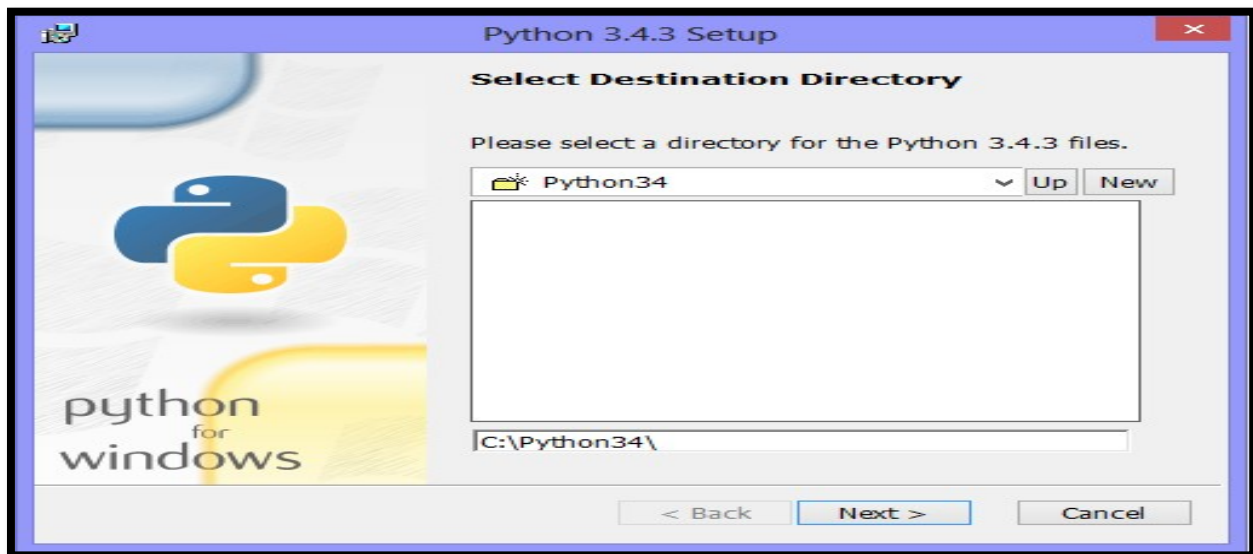
### 1. Double Click on The Python 3.4.3



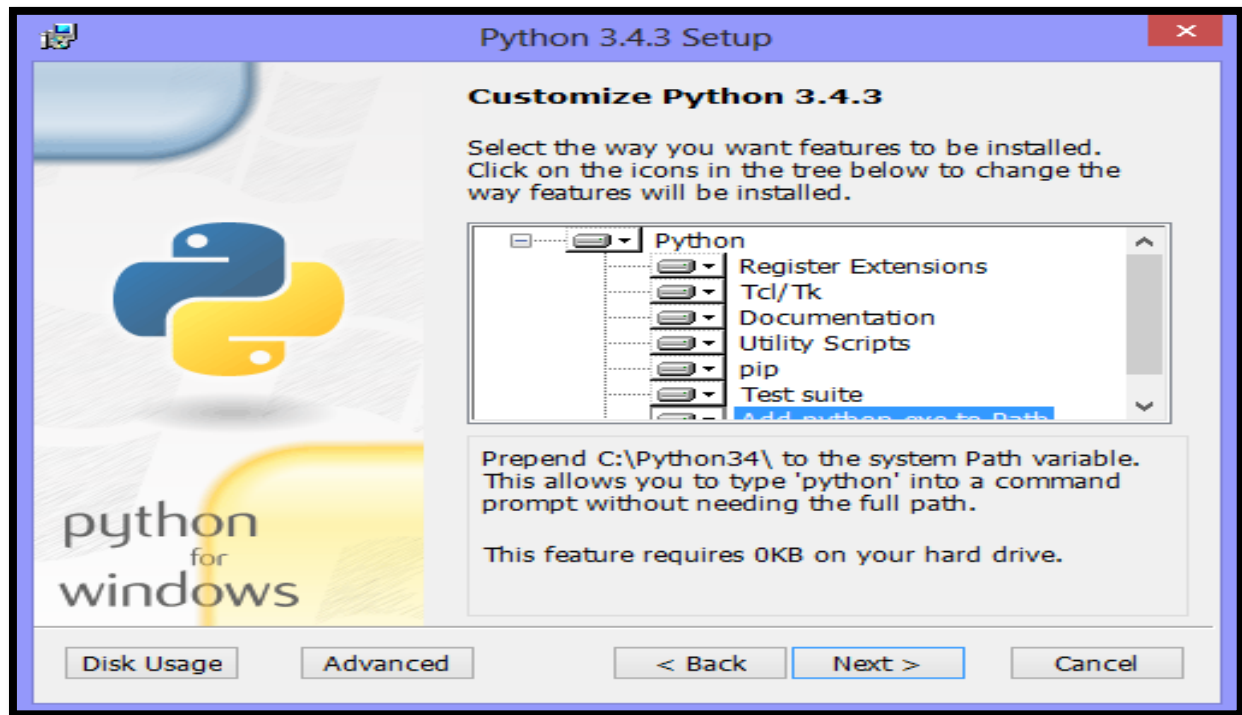
### 2. Click Next



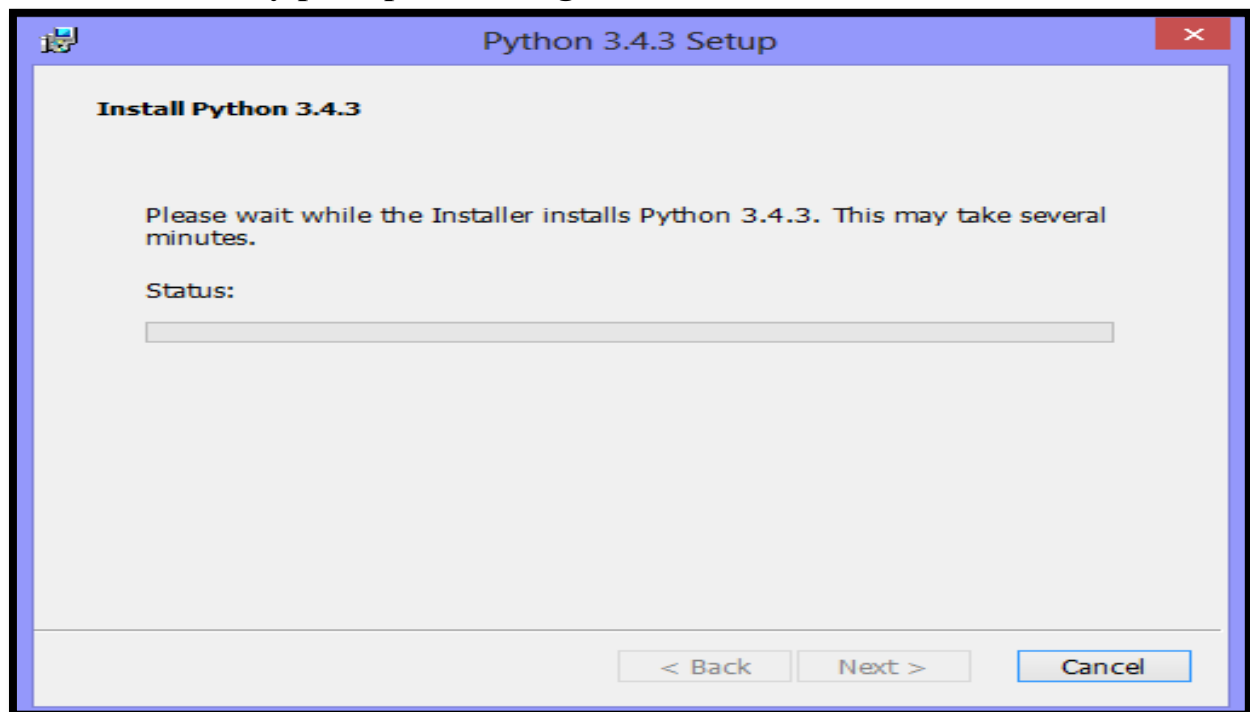
### 3. Click Next



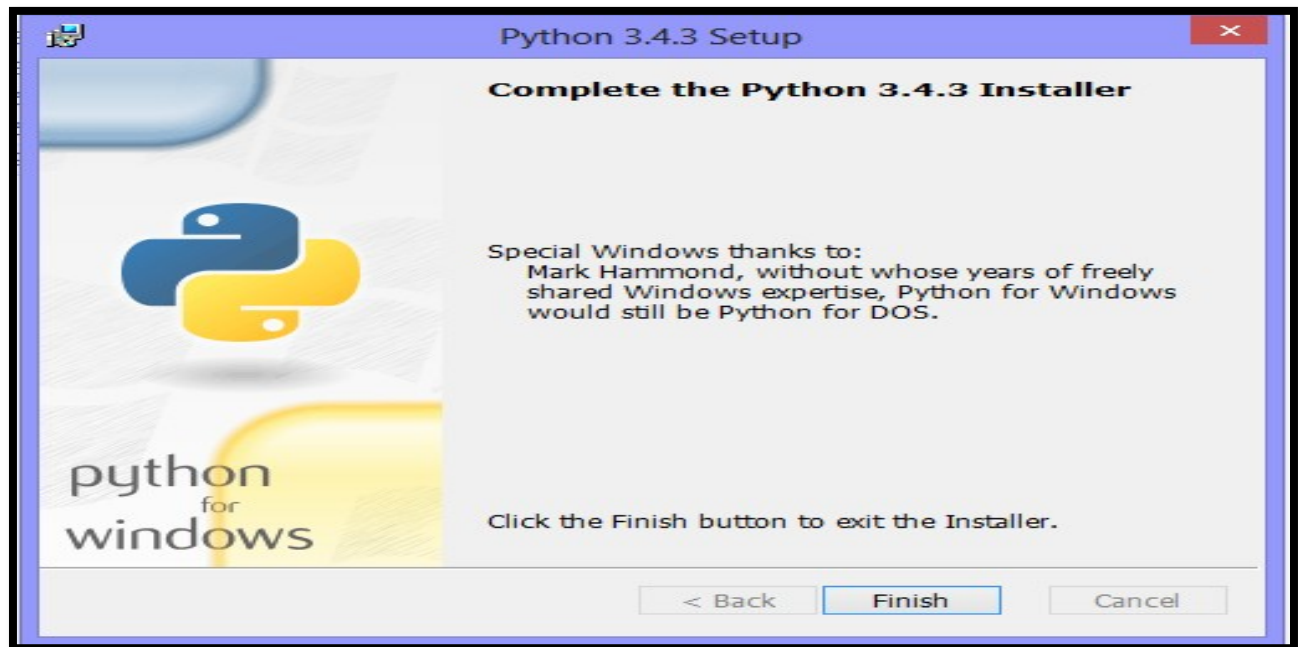
#### 4. Click Next



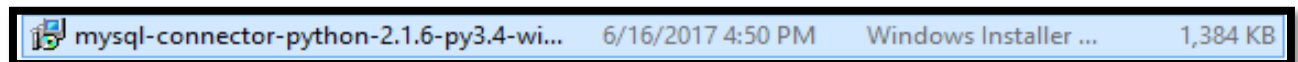
#### 5. Click Yes if any prompt is coming then click Next



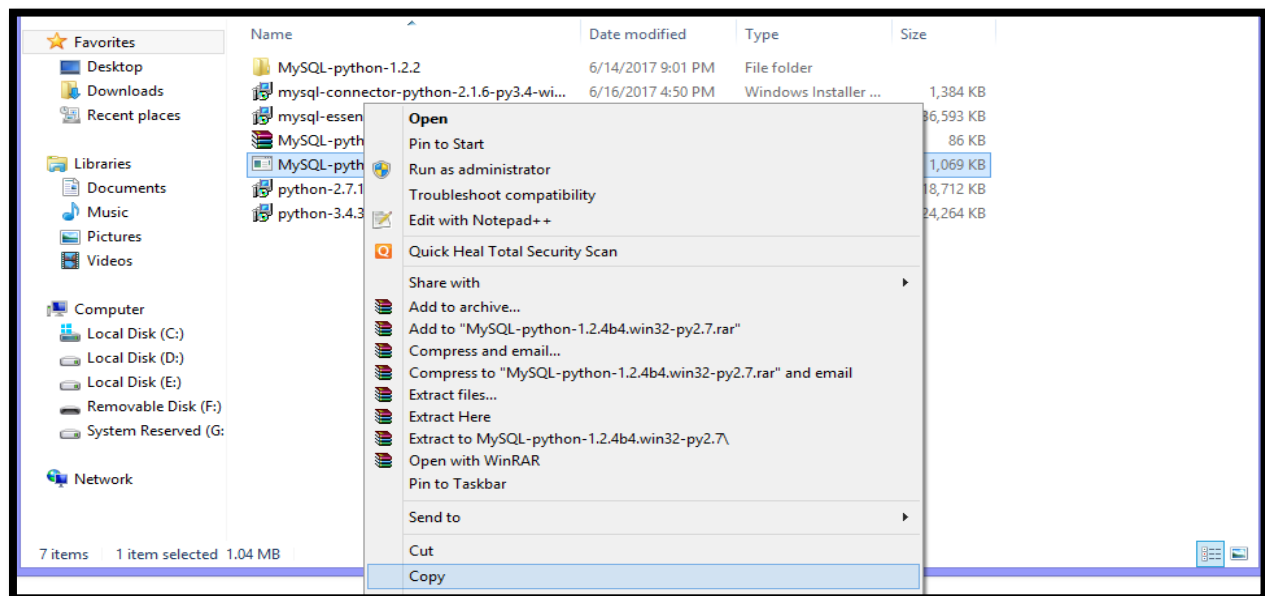
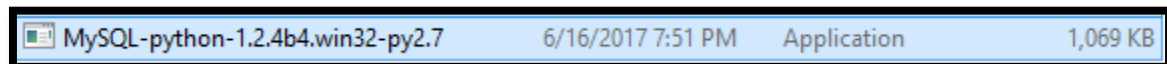
## 6. Click Finish

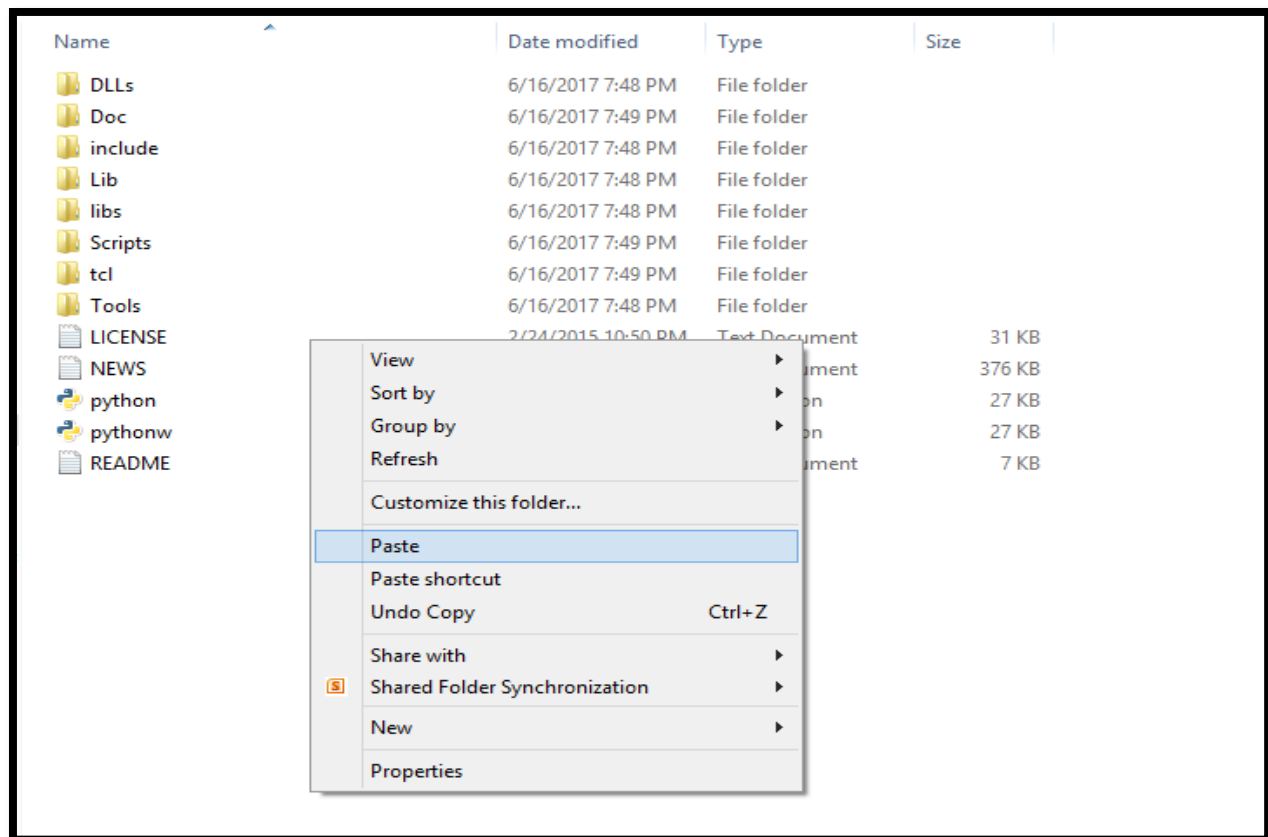
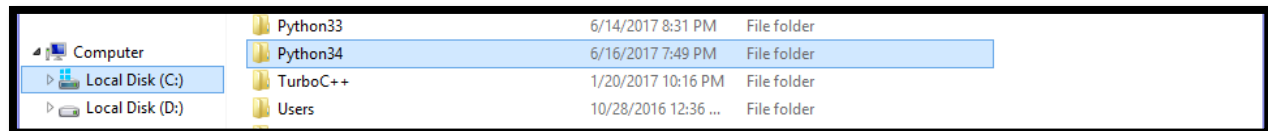


## 7. Now Install Mysql Connector for Python 3.4. Click on the Following Windows Installer File.

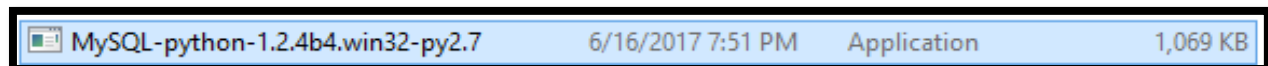


## 8. Now Copy the Following file in Python Directory.

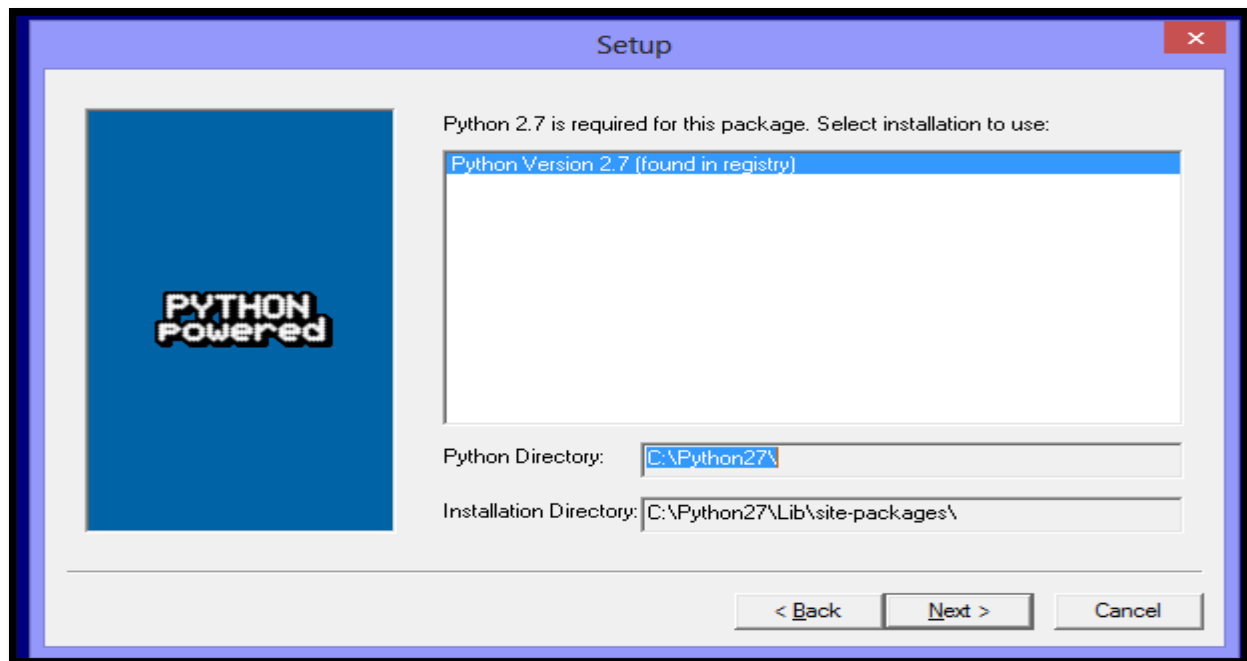
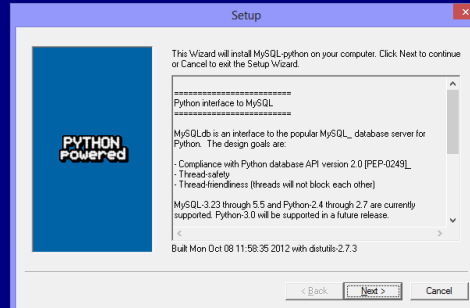


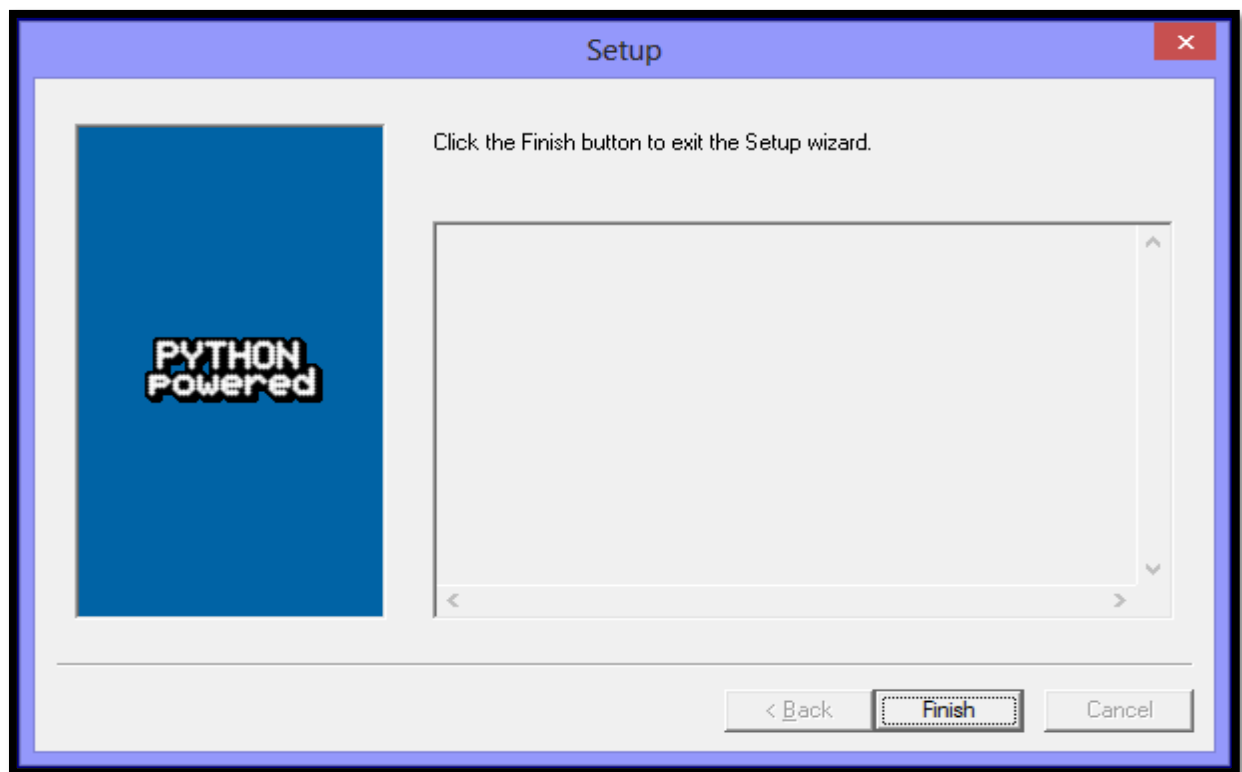
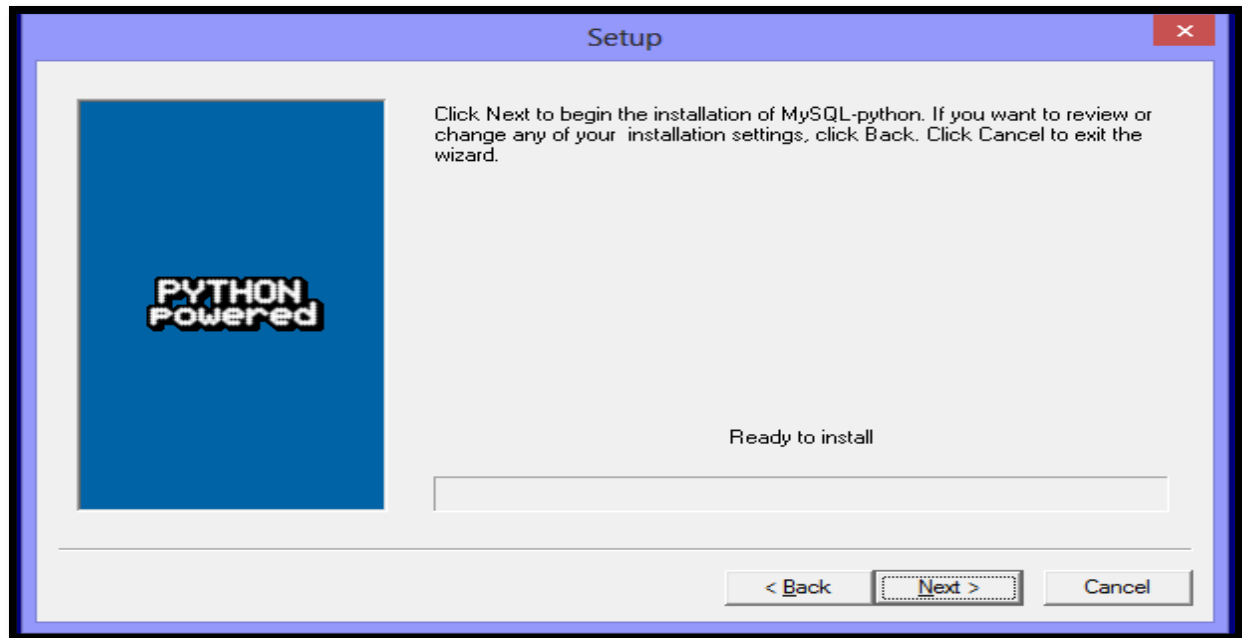


9. Now Double Click on this Application and follow the step.



# MySQL-python-1.2.4b4





**10. Now Check the MySQL Connectivity is done or not by Opening the IDLE (Python 3.4 GUI).**



### TestDB.py

```
import mysql.connector
db=mysql.connector.connect(user='root',passwd='root',host='127.0.0.1',database='nit')
# prepare a cursor object using cursor() method
cursor = db.cursor()
# execute SQL query using execute() method.
cursor.execute("SELECT VERSION()")
# Fetch a single row using fetchone() method.
data = cursor.fetchone()
print ("Database version : %s " % data)
# disconnect from server
db.close()
```

```
CheckDb.py - C:/Users/Nitesh/Desktop/PYTHON WORKSHOP MATERIAL/SYIT PYTHON PRAX CODE/CheckDb.py (3.4.3)
File Edit Format Run Options Window Help
import mysql.connector
db=mysql.connector.connect(user='root',passwd='root',host='127.0.0.1',database='nit')
# prepare a cursor object using cursor() method
cursor = db.cursor()
# execute SQL query using execute() method.
cursor.execute("SELECT VERSION()")
# Fetch a single row using fetchone() method.
data = cursor.fetchone()
print ("Database version : %s " % data)
# disconnect from server
db.close()
```

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Database version : 5.1.36-community
>>>
```

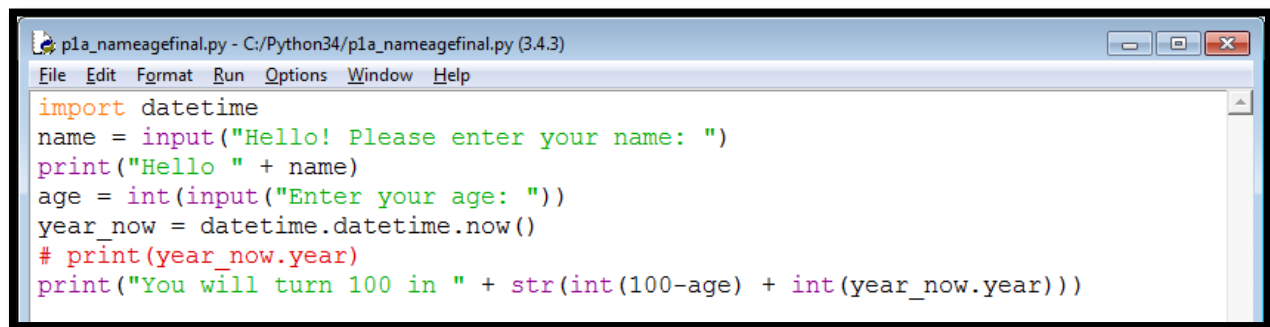


## Practical No.1

**1. Write the program for the following: (by using control statements and control structure)**

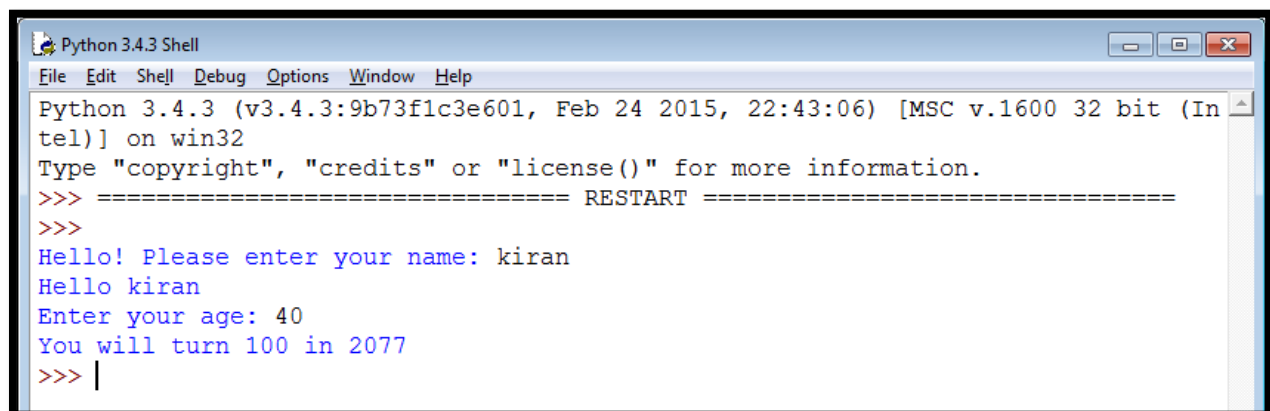
**A. Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old.**

```
import datetime
name = input("Hello! Please enter your name: ")
print("Hello " + name)
age = int(input("Enter your age: "))
year_now = datetime.datetime.now()
# print(year_now.year)
print("You will turn 100 in " + str(int(100-age) + int(year_now.year)))
```

A screenshot of a Python IDE window titled 'p1a\_nameagefinal.py - C:/Python34/p1a\_nameagefinal.py (3.4.3)'. The window contains the following code:

```
import datetime
name = input("Hello! Please enter your name: ")
print("Hello " + name)
age = int(input("Enter your age: "))
year_now = datetime.datetime.now()
# print(year_now.year)
print("You will turn 100 in " + str(int(100-age) + int(year_now.year)))
```

### **Output**

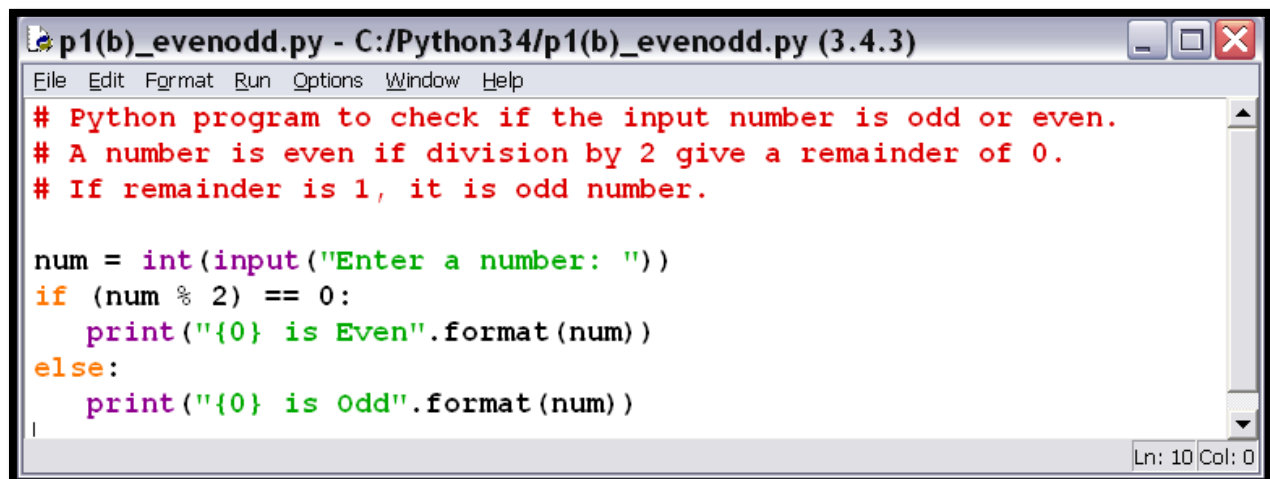
A screenshot of a Python 3.4.3 Shell window. The output shows the program's execution with user input 'kiran' and age '40', resulting in the message 'You will turn 100 in 2077'.

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Hello! Please enter your name: kiran
Hello kiran
Enter your age: 40
You will turn 100 in 2077
>>> |
```

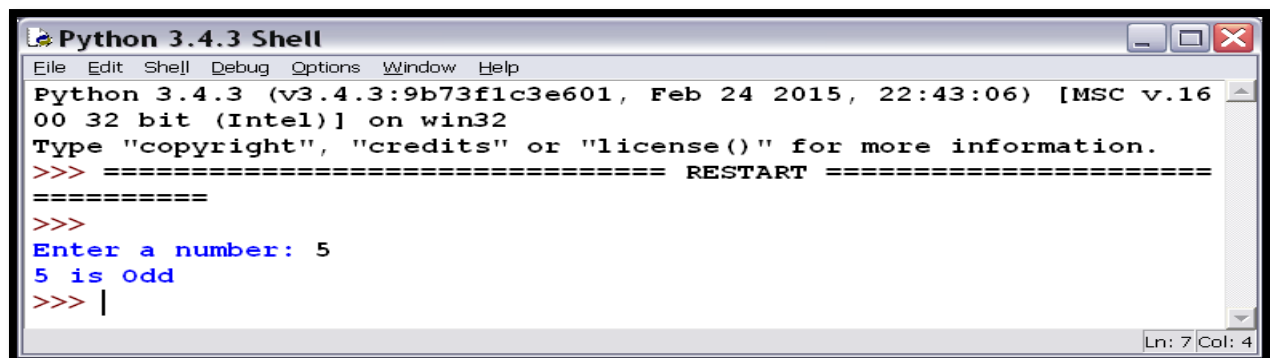
**B. Enter the number from the user and depending on whether the number is even or odd, print out an appropriate message to the user.**

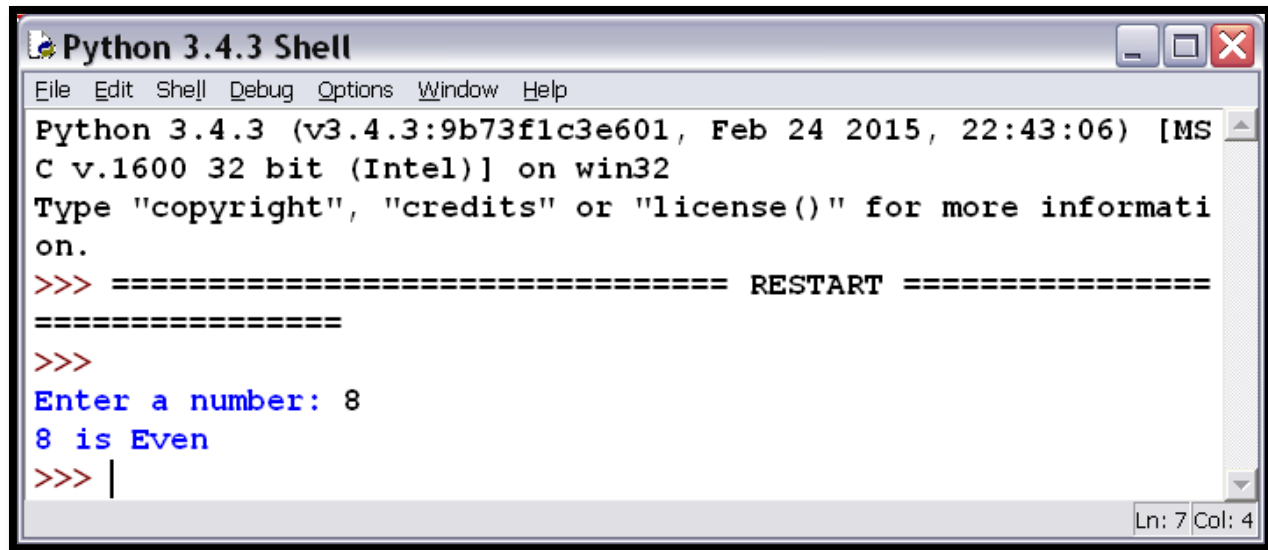
Code :

```
# Python program to check if the input number is odd or even.
# A number is even if division by 2 give a remainder of 0.
# If remainder is 1, it is odd number.
num = int(input("Enter a number: "))
if (num % 2) == 0:
    print("{0} is Even".format(num))
else:
    print("{0} is Odd".format(num))
Python Code :-
```



Output:-





```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MS
C v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more informati
on.
>>> ===== RESTART =====
>>>
Enter a number: 8
8 is Even
>>> |
```

### C. Write a program to generate the Fibonacci series.

# Program to display the Fibonacci sequence up to n-th term where n is provided by the user

# change this value for a different result

nterms = 10

# uncomment to take input from the user

#nterms = int(input("How many terms? "))

# first two terms

n1 = 0

n2 = 1

count = 2

# check if the number of terms is valid

if nterms <= 0:

    print("Please enter a positive integer")

elif nterms == 1:

    print("Fibonacci sequence upto", nterms, ":")

    print(n1)

else:

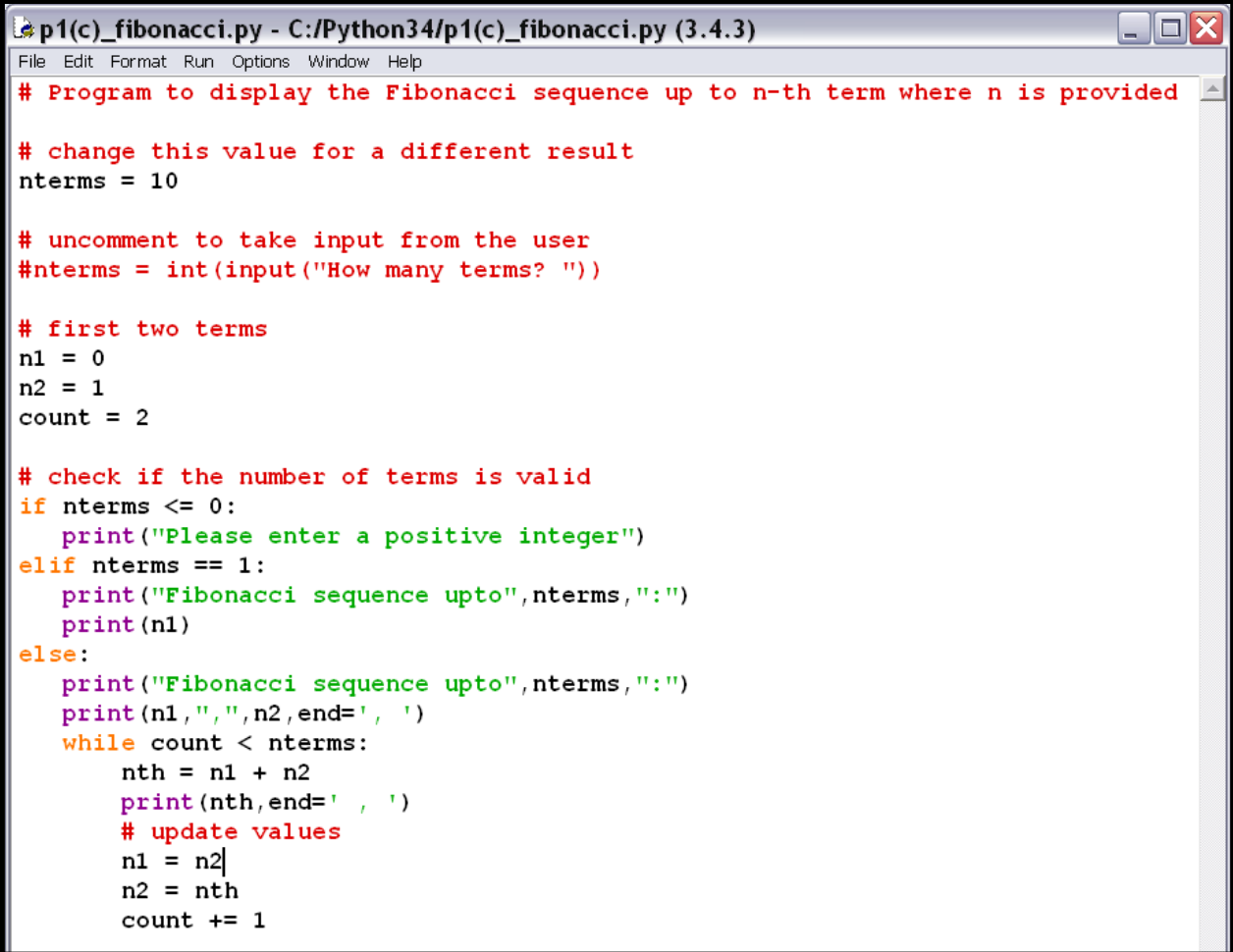
    print("Fibonacci sequence upto", nterms, ":")

    print(n1, ", ", n2, end=', ')

    while count < nterms:

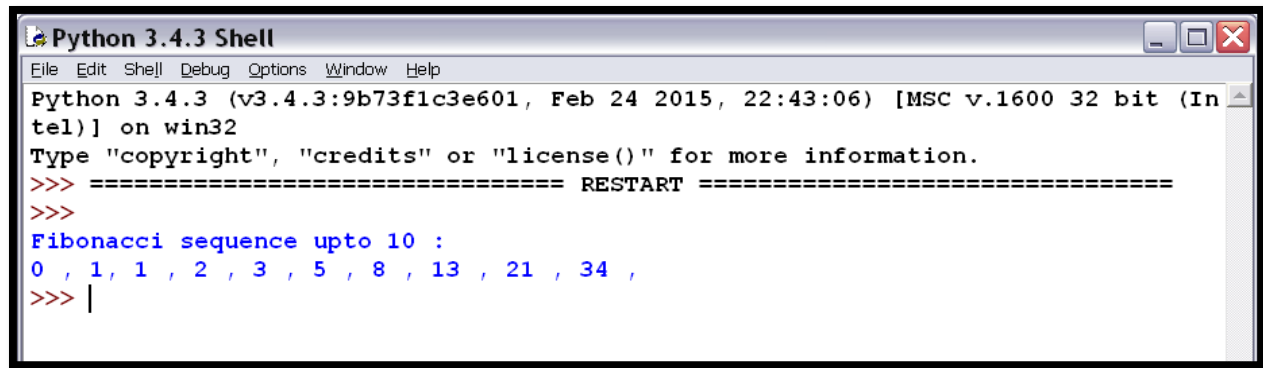
        nth = n1 + n2

```
print(nth,end=' , ')\n# update values\nn1 = n2\nn2 = nth\ncount += 1
```



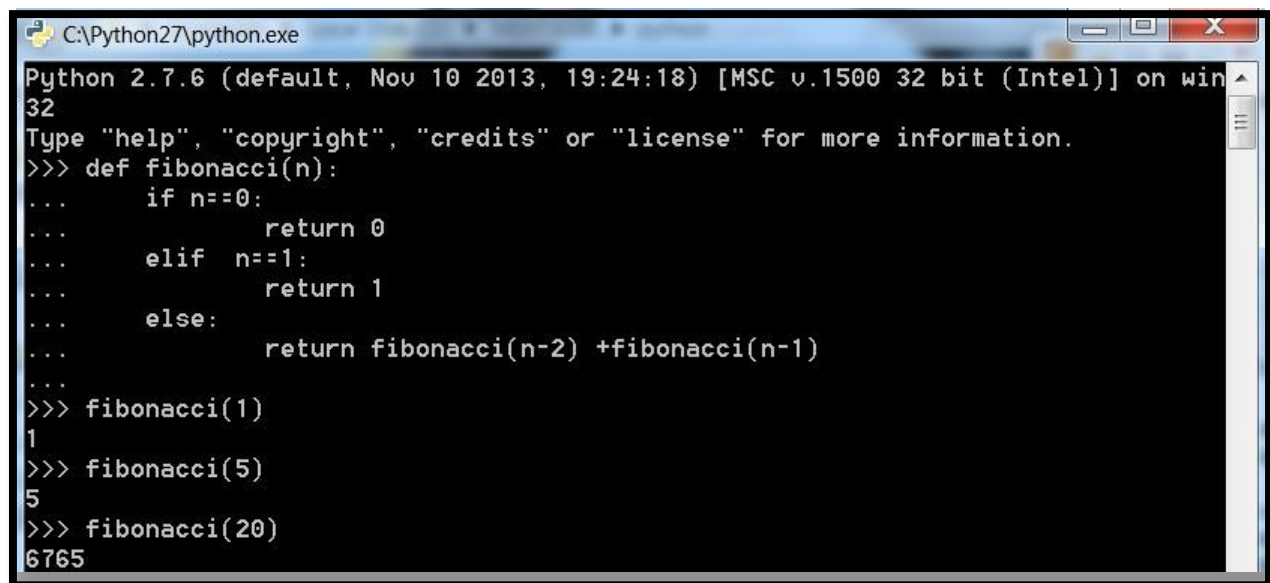
```
p1(c)_fibonacci.py - C:/Python34/p1(c)_fibonacci.py (3.4.3)\nFile Edit Format Run Options Window Help\n# Program to display the Fibonacci sequence up to n-th term where n is provided\n\n# change this value for a different result\nnterms = 10\n\n# uncomment to take input from the user\n#nterms = int(input("How many terms? "))\n\n# first two terms\nn1 = 0\nn2 = 1\ncount = 2\n\n# check if the number of terms is valid\nif nterms <= 0:\n    print("Please enter a positive integer")\nelif nterms == 1:\n    print("Fibonacci sequence upto",nterms,":")\n    print(n1)\nelse:\n    print("Fibonacci sequence upto",nterms,":")\n    print(n1, ", ", n2, end=' , ')\n    while count < nterms:\n        nth = n1 + n2\n        print(nth,end=' , ')\n        # update values\n        n1 = n2\n        n2 = nth\n        count += 1
```

## Output



```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Fibonacci sequence upto 10 :
0 , 1 , 1 , 2 , 3 , 5 , 8 , 13 , 21 , 34 ,
>>> |
```

### Fibonacci series by using function



```
C:\Python27\python.exe
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> def fibonacci(n):
...     if n==0:
...         return 0
...     elif n==1:
...         return 1
...     else:
...         return fibonacci(n-2) + fibonacci(n-1)
...
>>> fibonacci(1)
1
>>> fibonacci(5)
5
>>> fibonacci(20)
6765
```

### **D. Write a function that reverses the user defined value.**

# Python Program to Reverse a Number using While loop by using function

def reverse\_number(number):

    reverse = 0

    while(number > 0):

        remainder = number % 10

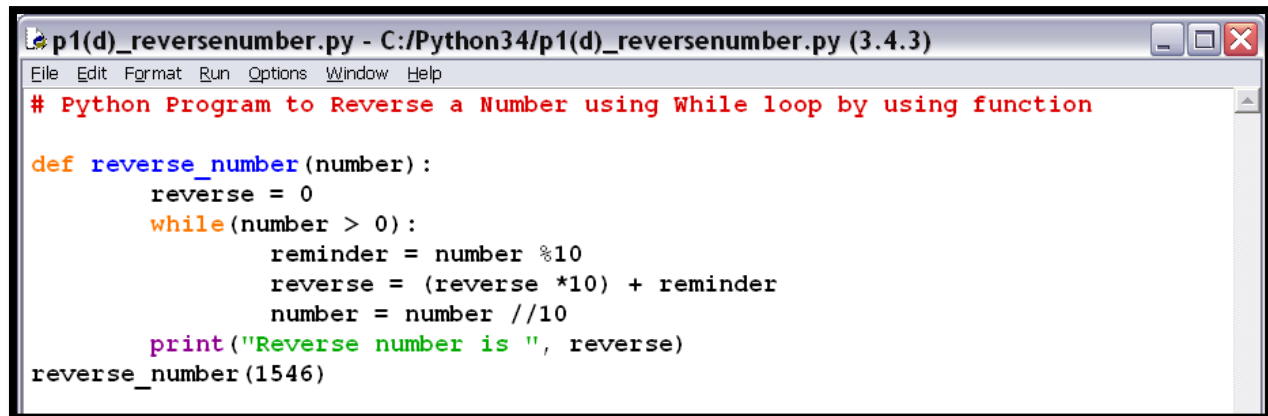
        reverse = (reverse \* 10) + remainder

        number = number // 10

    print("Reverse number is ", reverse)

reverse\_number(1546)

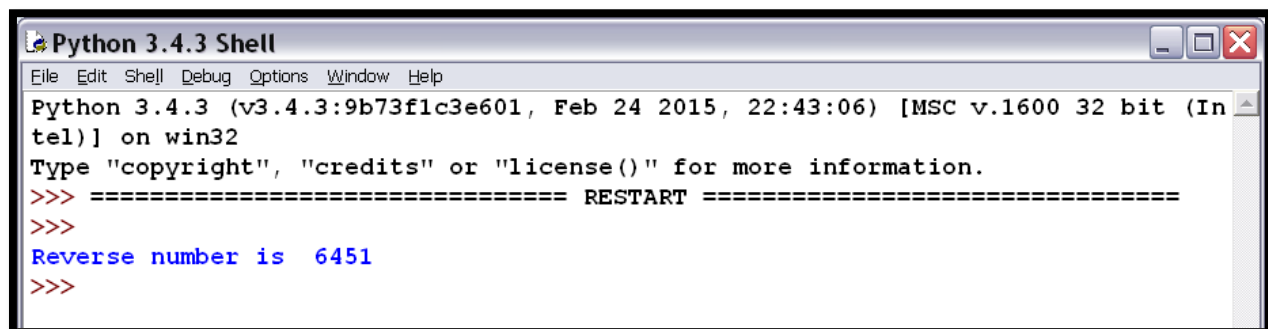
## Python code :



```
p1(d)_reversenumber.py - C:/Python34/p1(d)_reversenumber.py (3.4.3)
File Edit Format Run Options Window Help
# Python Program to Reverse a Number using While loop by using function

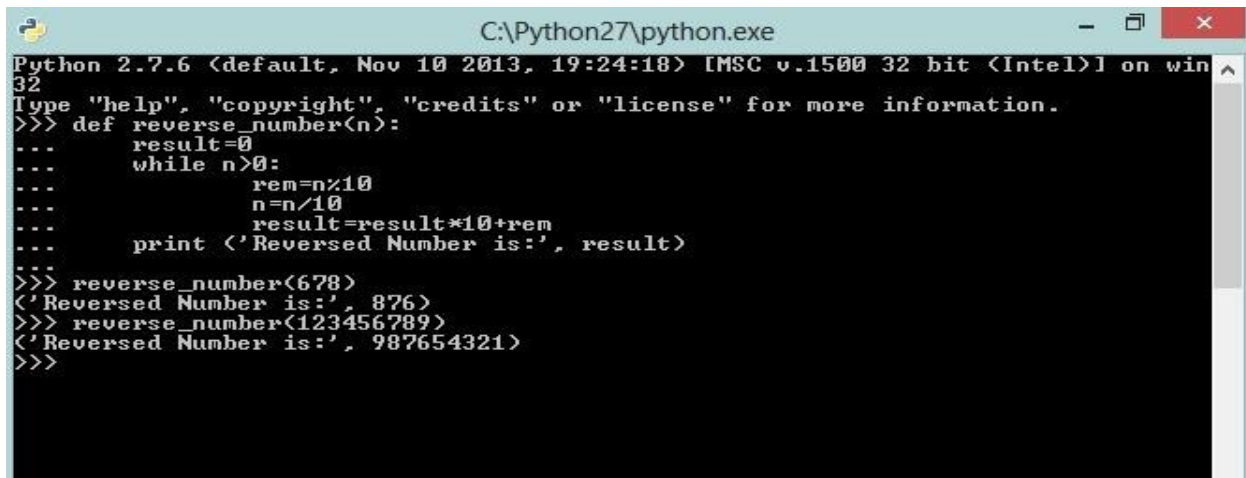
def reverse_number(number):
    reverse = 0
    while(number > 0):
        reminder = number %10
        reverse = (reverse *10) + reminder
        number = number //10
    print("Reverse number is ", reverse)
reverse_number(1546)
```

## Output



```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Reverse number is  6451
>>>
```

## Same Program on Python2.7 on Command prompt



```
C:\Python27\python.exe
Python 2.7.6 <default, Nov 10 2013, 19:24:18> [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> def reverse_number(n):
...     result=0
...     while n>0:
...         rem=n%10
...         n=n/10
...         result=result*10+rem
...     print <'Reversed Number is:', result>
...
>>> reverse_number(678)
<'Reversed Number is:', 876>
>>> reverse_number(123456789)
<'Reversed Number is:', 987654321>
>>>
```

**E. Write a function to check the input value is Armstrong and also write the function for Palindrome.**

Code:

# Python program to check if the number provided by the user is an Armstrong number or not

```
def armstrong(num):
    sum=0
    # find the sum of the cube of each digit
    temp = num
    while temp > 0:
        digit = temp % 10
        sum += digit ** 3
        temp //= 10
    # display the result
    if num == sum:
        print(num,"is an Armstrong number")
    else:
        print(num,"is not an Armstrong number")
```

```
def palindrome(num):
    n = num
    rev = 0
    while num != 0:
        rev = rev * 10
        rev = rev + int(num%10)
    num = int(num / 10)
    if n == rev:
        print(n,"is palindrome number")
    else:
        print(n,"is not a palin")
```

```
# take input from the user
num = int(input("Enter a number to chk it is armstrong or not: "))
armstrong(num)
# take input from the user
num = int(input("Enter a number to chk it is palindrome or not: "))
palindrome(num)
```

```
"ple_armstrong.py - C:/Python34/ple_armstrong.py (3.4.3)"
File Edit Format Run Options Window Help
# Python program to check if the number provided by the user is an Armstrong number or not
def armstrong(num):
    sum=0
    # find the sum of the cube of each digit
    temp = num
    while temp > 0:
        digit = temp % 10
        sum += digit ** 3
        temp //= 10
    # display the result
    if num == sum:
        print(num,"is an Armstrong number")
    else:
        print(num,"is not an Armstrong number")

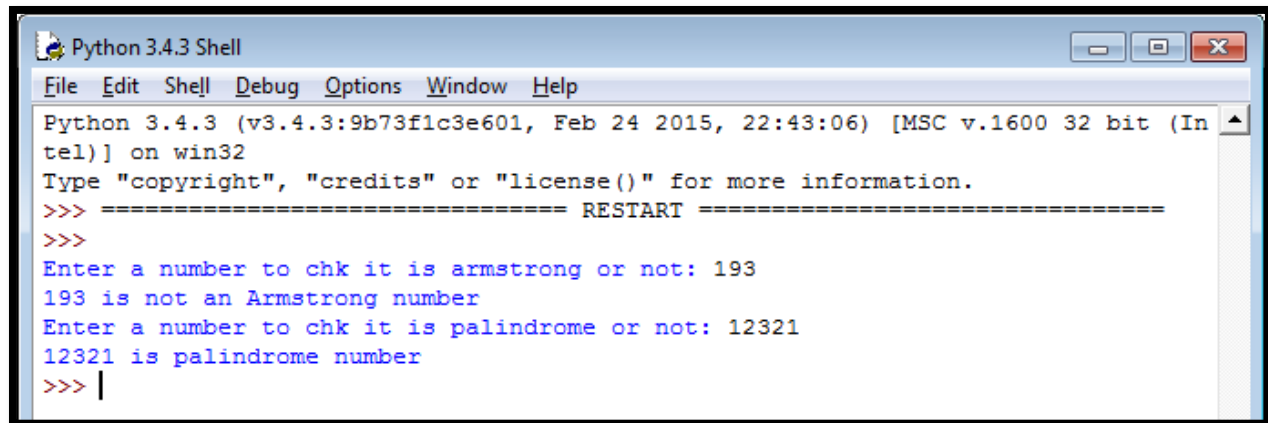
def palindrome (num):
    n = num
    rev = 0
    while num != 0:
        rev = rev * 10
        rev = rev + int(num%10)
        num = int(num / 10)
    if n == rev:
        print(n,"is palindrome number")
    else:
        print(n,"is not a palin")

# take input from the user
num = int(input("Enter a number to chk it is armstrong or not: "))
armstrong(num)
# take input from the user
num = int(input("Enter a number to chk it is palindrome or not: "))
palindrome(num)
```

## Output

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Enter a number to chk it is armstrong or not: 153
153 is an Armstrong number
Enter a number to chk it is palindrome or not: 1232
1232 is not a palin
>>>
```





```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Enter a number to chk it is armstrong or not: 193
193 is not an Armstrong number
Enter a number to chk it is palindrome or not: 12321
12321 is palindrome number
>>> |
```

## F. Write a recursive function to print the factorial for a given number.

# Python program to find the factorial of a number using recursion

```
def recur_factorial(n):
```

```
    """Function to return the factorial
    of a number using recursion"""
```

```
    if n == 1:
```

```
        return n
```

```
    else:
```

```
        return n*recur_factorial(n-1)
```

```
#take input from the user
```

```
num = int(input("Enter a number: "))
```

```
# check is the number is negative
```

```
if num < 0:
```

```
    print("Sorry, factorial does not exist for negative numbers")
```

```
elif num == 0:
```

```
    print("The factorial of 0 is 1")
```

```
else:
```

```
    print("The factorial of",num,"is",recur_factorial(num))
```

```
p1f_recursionfact.py - C:/Python34/p1f_recursionfact.py (3.4.3)
File Edit Format Run Options Window Help
# Python program to find the factorial of a number using recursion

def recur_factorial(n):
    """Function to return the factorial
    of a number using recursion"""
    if n == 1:
        return n
    else:
        return n*recur_factorial(n-1)

#take input from the user
num = int(input("Enter a number: "))

# check is the number is negative
if num < 0:
    print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
    print("The factorial of 0 is 1")
else:
    print("The factorial of",num,"is",recur_factorial(num))
```

## Output

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Enter a number: 5
The factorial of 5 is 120
>>>
```

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Enter a number: -5
Sorry, factorial does not exist for negative numbers
>>>
```

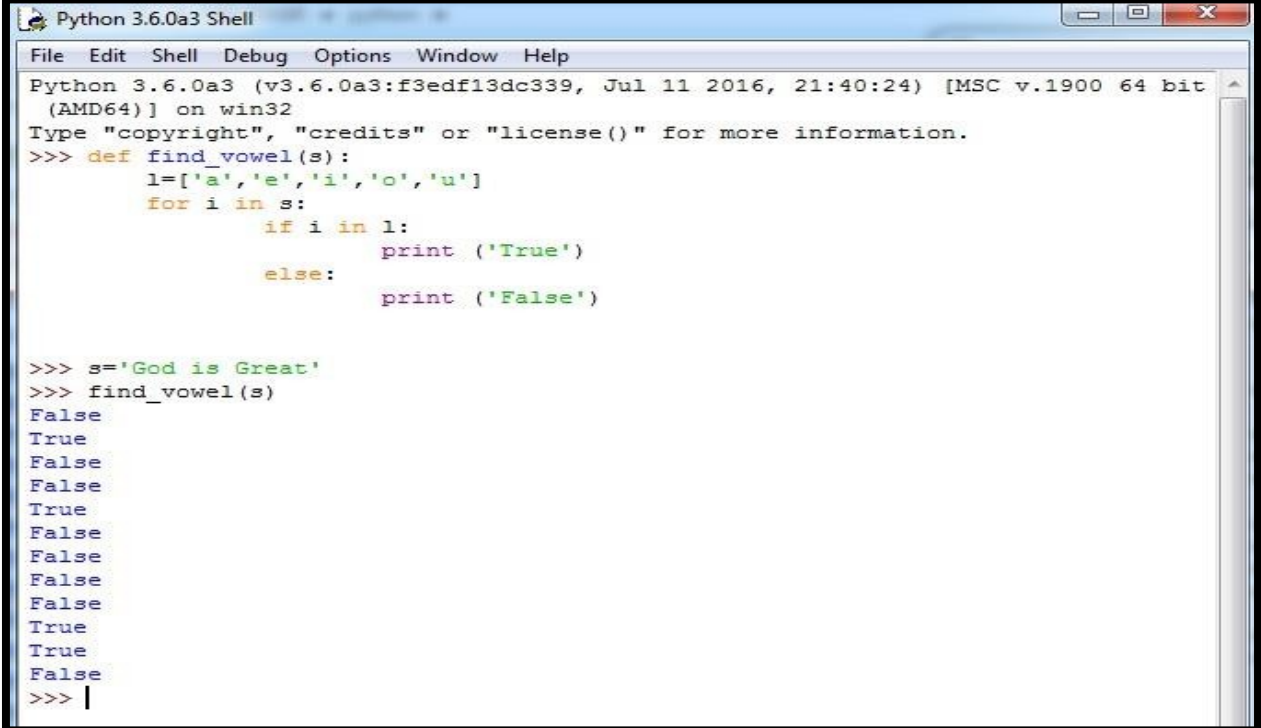
```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Enter a number: 0
The factorial of 0 is 1
>>>
```

## Practical No.2

**Write the program for the following: ( by using functions)**

**A. Write a function that takes a character (i.e. a string of length 1) and returns True if it is a vowel, False otherwise.**

The code and the output is shown in the following screenshot.

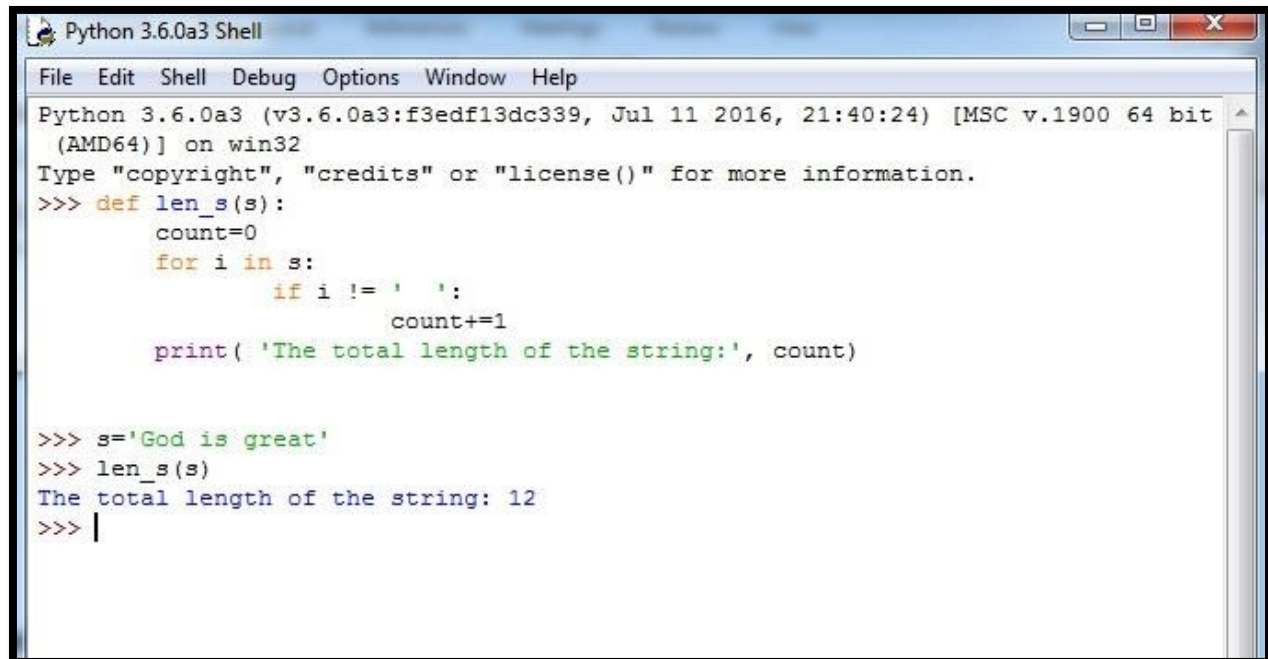
A screenshot of a Python 3.6.0a3 Shell window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The title bar says 'Python 3.6.0a3 Shell'. The main text area shows the following code and output:

```
Python 3.6.0a3 (v3.6.0a3:f3edf13dc339, Jul 11 2016, 21:40:24) [MSC v.1900 64 bit
(AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> def find_vowel(s):
    l=['a','e','i','o','u']
    for i in s:
        if i in l:
            print ('True')
        else:
            print ('False')

>>> s='God is Great'
>>> find_vowel(s)
False
True
False
False
True
False
False
False
False
True
True
False
>>> |
```

**B. Define a function that computes the length of a given list or string.**

The code and the output is shown in the following screenshot.



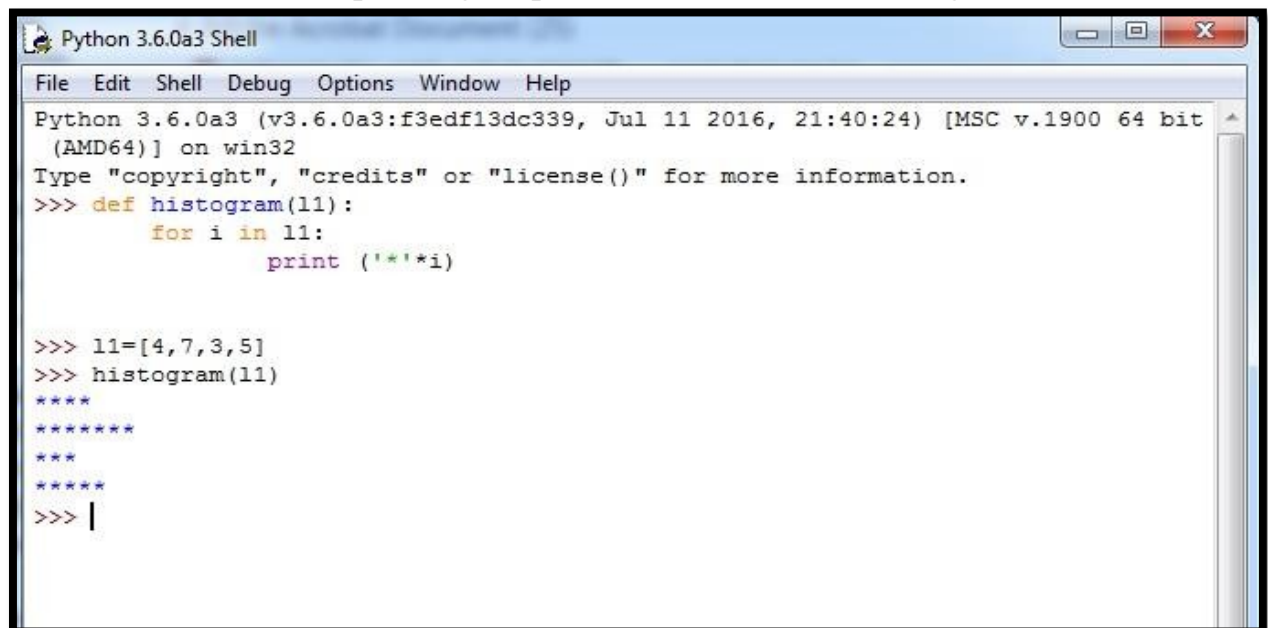
```
Python 3.6.0a3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0a3 (v3.6.0a3:f3edf13dc339, Jul 11 2016, 21:40:24) [MSC v.1900 64 bit
(AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> def len_s(s):
    count=0
    for i in s:
        if i != ' ':
            count+=1
    print( 'The total length of the string:', count)

>>> s='God is great'
>>> len_s(s)
The total length of the string: 12
>>> |
```

**C. Define a procedure histogram() that takes a list of integers and prints a histogram to the screen. For example, histogram([4, 9, 7]) should print the following:**

```
****
*****
** *
*****
```

The code and the corresponding output is shown in the following screen shot.



```
Python 3.6.0a3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0a3 (v3.6.0a3:f3edf13dc339, Jul 11 2016, 21:40:24) [MSC v.1900 64 bit
(AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> def histogram(l1):
    for i in l1:
        print ('*' * i)

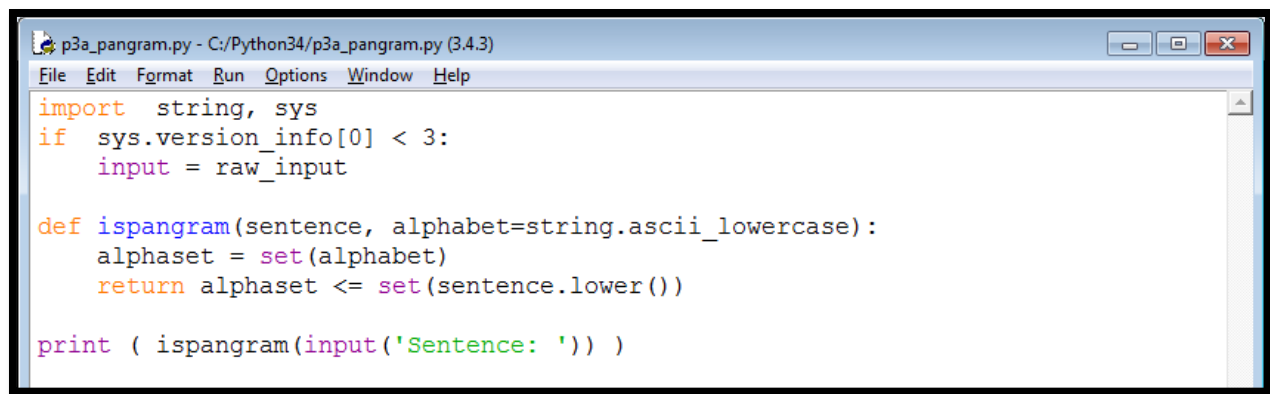
>>> l1=[4,7,3,5]
>>> histogram(l1)
****
*****
***
*****
>>> |
```

## Practical No.-3

**Write the program for the following: ( by using list)**

**A. A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: The quick brown fox jumps over the lazy dog. Your task here is to write a function to check a sentence to see if it is a pangram or not.**

```
import string, sys
if sys.version_info[0] < 3:
    input = raw_input
def ispangram(sentence, alphabet=string.ascii_lowercase):
    alphasets = set(alphabet)
    return alphasets <= set(sentence.lower())
print ( ispangram(input('Sentence: ')) )
```

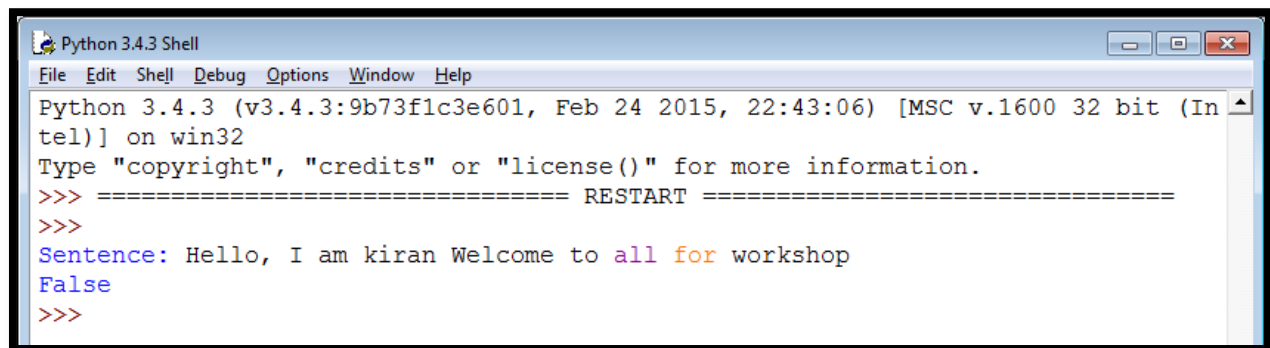
A screenshot of a Python IDE window titled 'p3a\_pangram.py - C:/Python34/p3a\_pangram.py (3.4.3)'. The window contains the following code:

```
import string, sys
if sys.version_info[0] < 3:
    input = raw_input

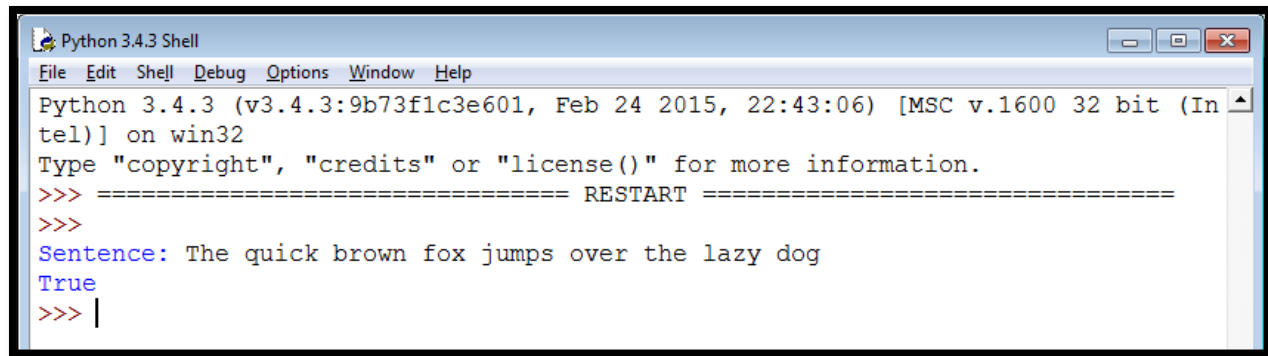
def ispangram(sentence, alphabet=string.ascii_lowercase):
    alphasets = set(alphabet)
    return alphasets <= set(sentence.lower())

print ( ispangram(input('Sentence: ')) )
```

## Output

A screenshot of a Python 3.4.3 Shell window. The window shows the following output:

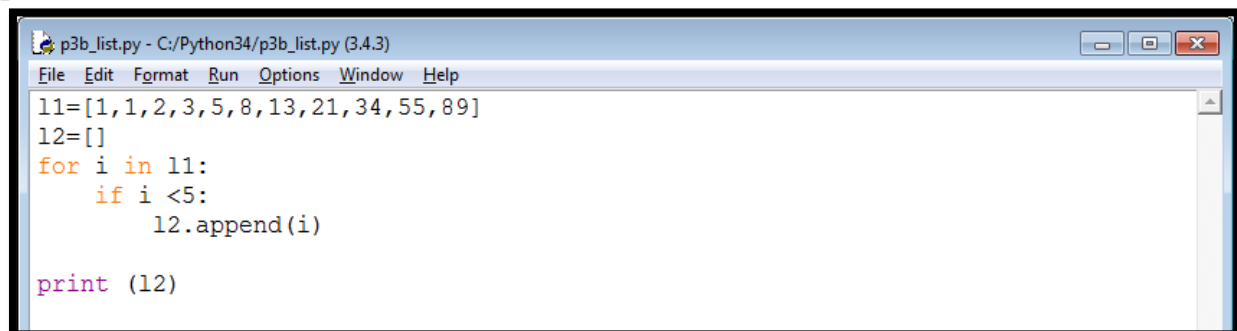
```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Sentence: Hello, I am kiran Welcome to all for workshop
False
>>>
```



```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Sentence: The quick brown fox jumps over the lazy dog
True
>>> |
```

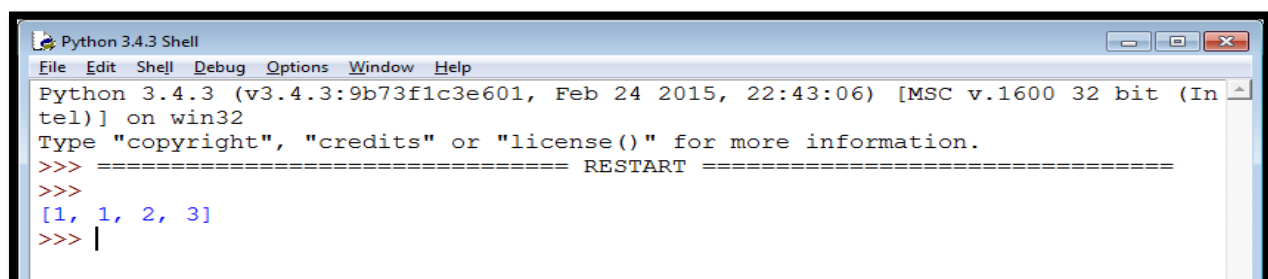
**B. Take a list, say for example this one: a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] and write a program that prints out all the elements of the list that are less than 5.**

```
# Program to trim list list_trim.py
l1=[1,1,2,3,5,8,13,21,34,55,89]
l2=[]
for i in l1:
    if i <5:
        l2.append(i)
print (l2)
```



```
p3b_list.py - C:/Python34/p3b_list.py (3.4.3)
File Edit Format Run Options Window Help
l1=[1,1,2,3,5,8,13,21,34,55,89]
l2=[]
for i in l1:
    if i <5:
        l2.append(i)

print (l2)
```



```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
[1, 1, 2, 3]
>>> |
```

## Practical No.4

**Write the program for the following: ( by using list)**

**A. Write a program that takes two lists and returns True if they have at least one common member.**

Code :

```
l1=[1,2,3,4,5,6,]
```

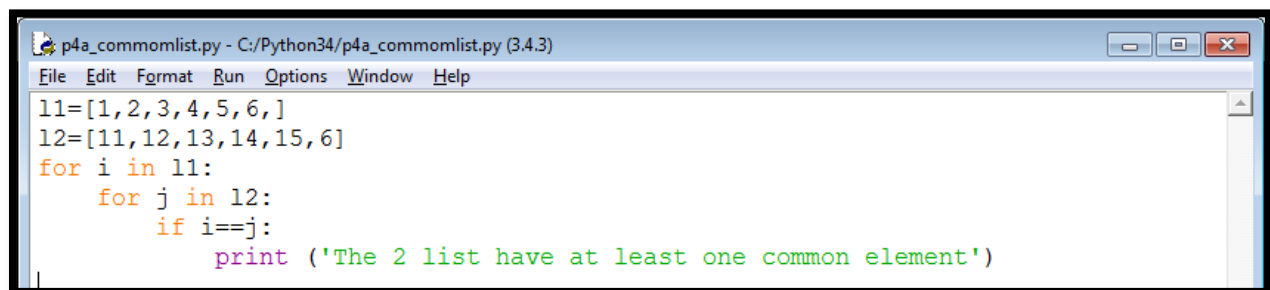
```
l2=[11,12,13,14,15,6]
```

```
for i in l1:
```

```
    for j in l2:
```

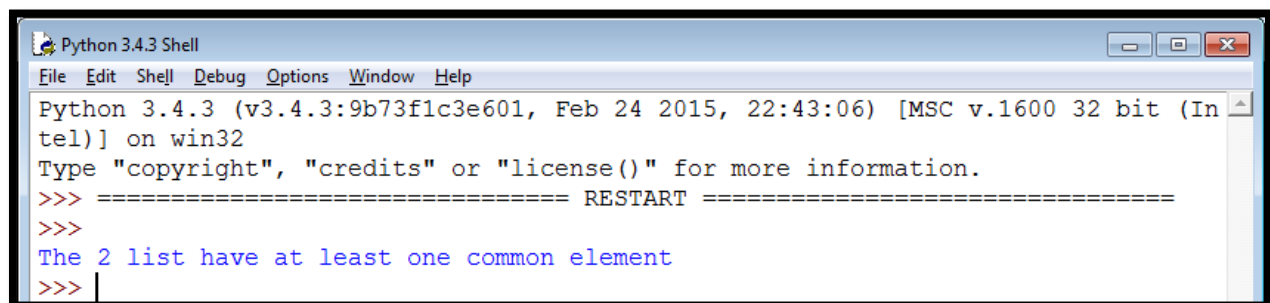
```
        if i==j:
```

```
            print ('The 2 list have at least one common element')
```



```
p4a_commomlist.py - C:/Python34/p4a_commomlist.py (3.4.3)
File Edit Format Run Options Window Help
l1=[1,2,3,4,5,6,]
l2=[11,12,13,14,15,6]
for i in l1:
    for j in l2:
        if i==j:
            print ('The 2 list have at least one common element')
```

### Output



```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
The 2 list have at least one common element
>>>
```

**B. Write a Python program to print a specified list after removing the 0th, 2nd, 4th and 5th elements.**

#print list after removing the 0th, 2nd, 4th and 5th elements.

```
l1=[1,2,3,4,5,6,7,8,9,0]
```

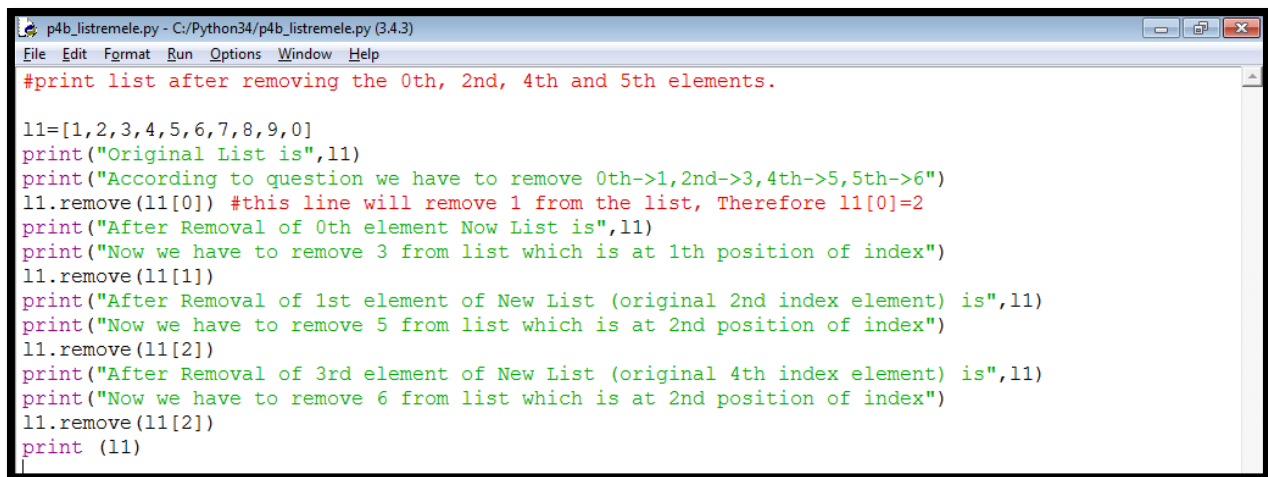
```
print("Original List is",l1)
```



```

print("According to question we have to remove 0th->1,2nd->3,4th->5,5th->6")
l1.remove(l1[0]) #this line will remove 1 from the list, Therefore l1[0]=2
print("After Removal of 0th element Now List is",l1)
print("Now we have to remove 3 from list which is at 1th position of index")
l1.remove(l1[1])
print("After Removal of 1st element of New List (original 2nd index element)
is",l1)
print("Now we have to remove 5 from list which is at 2nd position of index")
l1.remove(l1[2])
print("After Removal of 3rd element of New List (original 4th index element)
is",l1)
print("Now we have to remove 6 from list which is at 2nd position of index")
l1.remove(l1[2])
print (l1)

```



A screenshot of a Python script editor window titled 'p4b\_listremele.py - C:/Python34/p4b\_listremele.py (3.4.3)'. The code inside the editor is as follows:

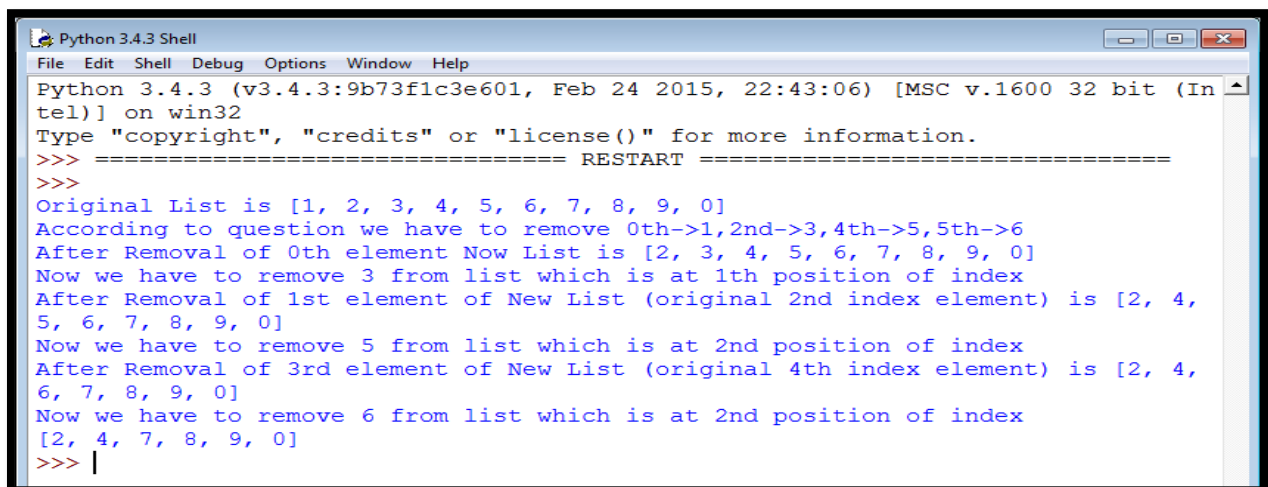
```

#print list after removing the 0th, 2nd, 4th and 5th elements.

l1=[1,2,3,4,5,6,7,8,9,0]
print("Original List is",l1)
print("According to question we have to remove 0th->1,2nd->3,4th->5,5th->6")
l1.remove(l1[0]) #this line will remove 1 from the list, Therefore l1[0]=2
print("After Removal of 0th element Now List is",l1)
print("Now we have to remove 3 from list which is at 1th position of index")
l1.remove(l1[1])
print("After Removal of 1st element of New List (original 2nd index element) is",l1)
print("Now we have to remove 5 from list which is at 2nd position of index")
l1.remove(l1[2])
print("After Removal of 3rd element of New List (original 4th index element) is",l1)
print("Now we have to remove 6 from list which is at 2nd position of index")
l1.remove(l1[2])
print (l1)

```

## Output



A screenshot of a Python 3.4.3 Shell window. The output of the script is displayed as follows:

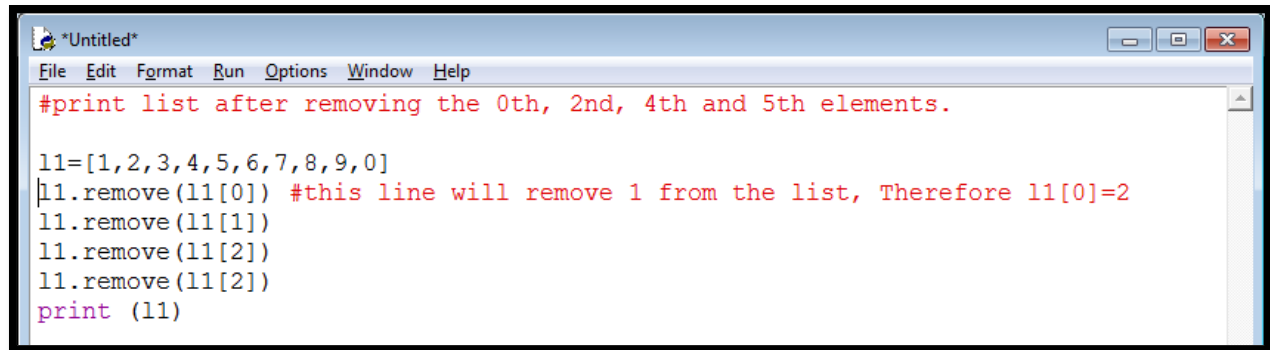
```

Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Original List is [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
According to question we have to remove 0th->1,2nd->3,4th->5,5th->6
After Removal of 0th element Now List is [2, 3, 4, 5, 6, 7, 8, 9, 0]
Now we have to remove 3 from list which is at 1th position of index
After Removal of 1st element of New List (original 2nd index element) is [2, 4, 5, 6, 7, 8, 9, 0]
Now we have to remove 5 from list which is at 2nd position of index
After Removal of 3rd element of New List (original 4th index element) is [2, 4, 6, 7, 8, 9, 0]
Now we have to remove 6 from list which is at 2nd position of index
[2, 4, 7, 8, 9, 0]
>>> |

```



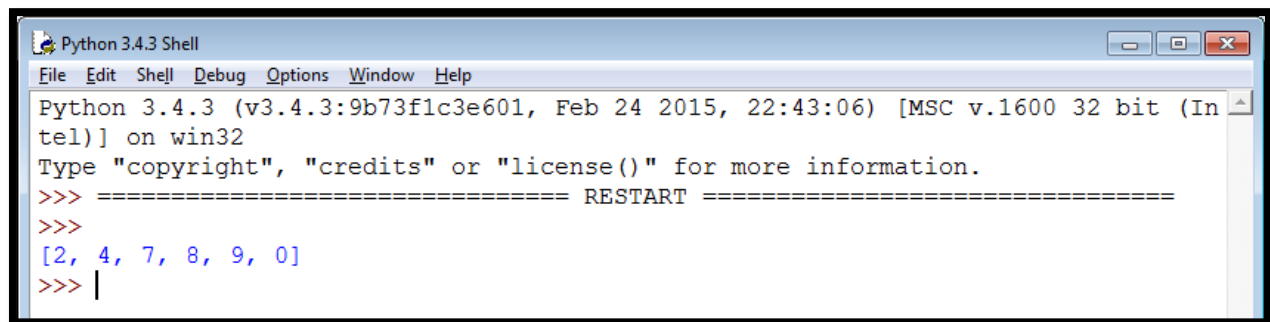
**You can try without print statements**



```
*Untitled*
File Edit Format Run Options Window Help
#print list after removing the 0th, 2nd, 4th and 5th elements.

l1=[1,2,3,4,5,6,7,8,9,0]
l1.remove(l1[0]) #this line will remove 1 from the list, Therefore l1[0]=2
l1.remove(l1[1])
l1.remove(l1[2])
l1.remove(l1[2])
print (l1)
```

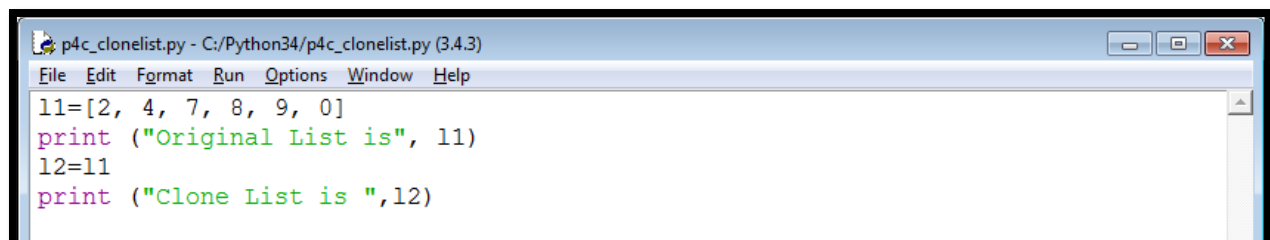
## Output



```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
[2, 4, 7, 8, 9, 0]
>>> |
```

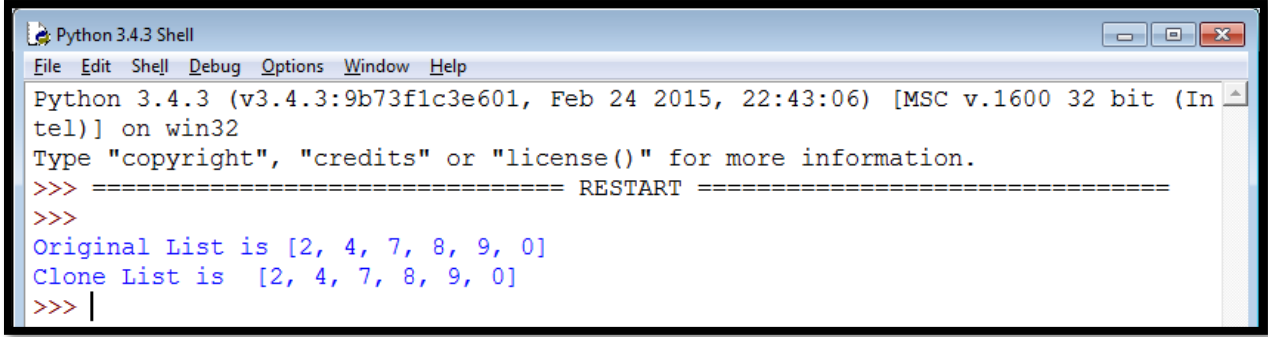
## **C. Write a Python program to clone or copy a list**

```
l1=[2, 4, 7, 8, 9, 0]
print ("Original List is", l1)
l2=l1
print ("Clone List is ",l2)
```



```
p4c_clonelist.py - C:/Python34/p4c_clonelist.py (3.4.3)
File Edit Format Run Options Window Help
l1=[2, 4, 7, 8, 9, 0]
print ("Original List is", l1)
l2=l1
print ("Clone List is ",l2)
```

## Output



```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Original List is [2, 4, 7, 8, 9, 0]
Clone List is [2, 4, 7, 8, 9, 0]
>>> |
```

## Practical No.5

**Write the program for the following: ( by using Dictionary)**

**A. Write a Python script to sort (ascending and descending) a dictionary by value.**

```
>>> released={'Python 3.6': 2017,'Python 1.0': 2002, 'Python 2.3': 2010}
>>> for key,value in sorted(released.items()):
    print (key,value)
```

**Output:**

Python 1.0 2002

Python 2.3 2010

Python 3.6 2017

Only keys sorted:

```
>>> print (sorted(released))
['Python 1.0', 'Python 2.3', 'Python 3.6']
```

**B. Write a Python script to concatenate following dictionaries to create a new one.**

Sample Dictionary : dic1={1:10, 2:20} dic2={3:30, 4:40} dic3={5:50,6:60}  
Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

**Output:**

```
>>> dic1={1:10,2:20}
>>> dic2={3:30,4:40}
>>> dic3={5:50,6:60}
>>> dic1.update(dic2)
>>> print (dic1)
{1: 10, 2: 20, 3: 30, 4: 40}
>>> dic1.update(dic3)
>>> print (dic1)

{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

**C. Write a Python program to sum all the items in a dictionary.**

```
>>> d= {'One':10,'Two':20,'Three':30}
```

```
>>> sum(d.values())
```

```
60
```

## Practical No.6

**Write the program for the following: ( File handling)**

**A. Write a Python program to read an entire text file.**

**Code:**

```
'''
```

Write a Python program to read an entire text file.

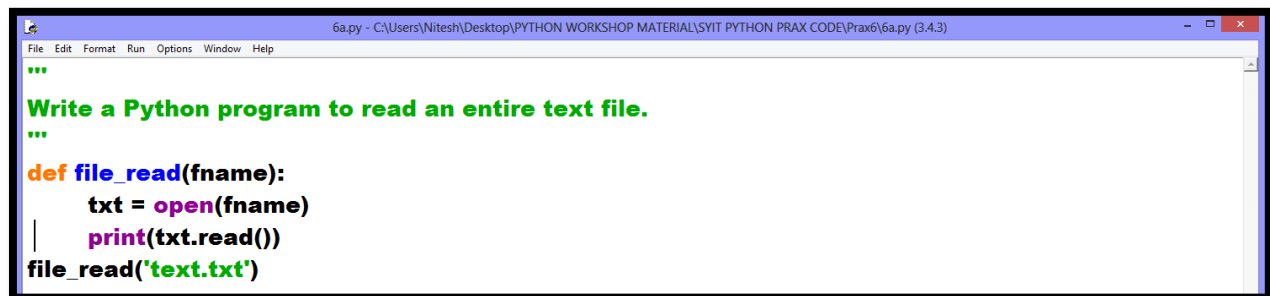
```
'''
```

```
def file_read(fname):
```

```
    txt = open(fname)
```

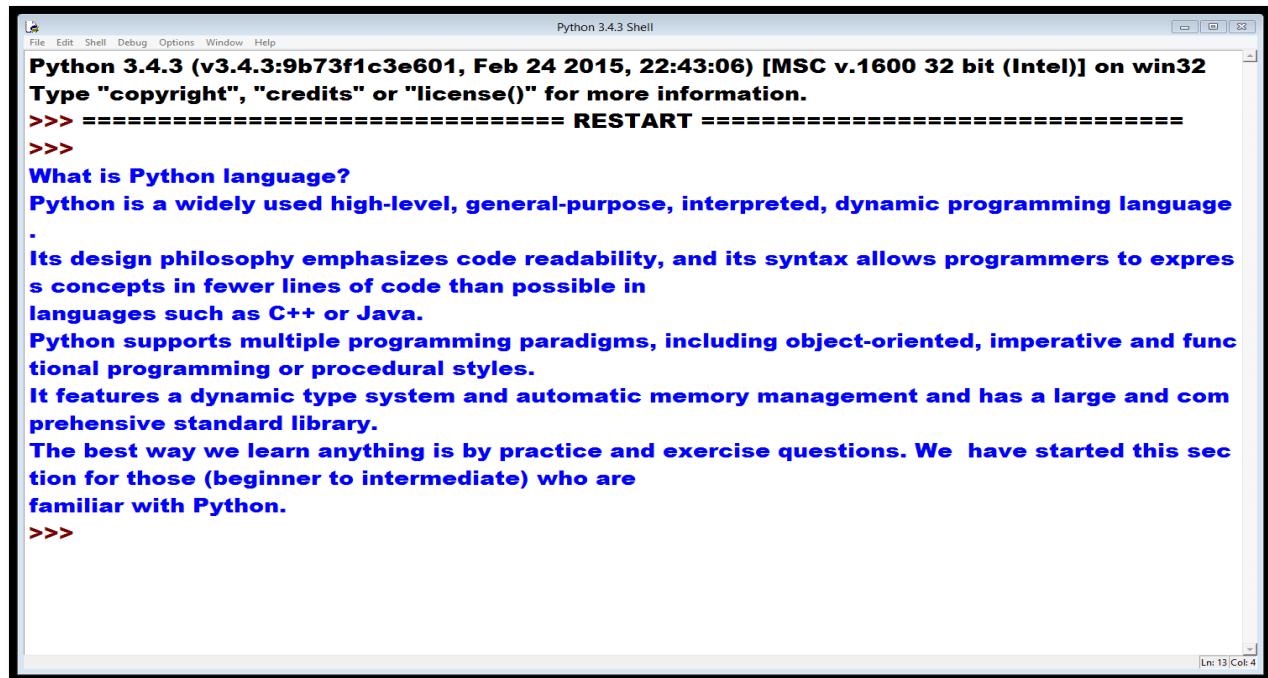
```
    print(txt.read())
```

```
file_read('text.txt')
```

A screenshot of a Python IDE window titled '6a.py - C:\Users\Nitesh\Desktop\PYTHON WORKSHOP MATERIAL\SYIT PYTHON PRAX CODE\Prax6\6a.py (3.4.3)'. The window contains the following code:

```
'''  
Write a Python program to read an entire text file.  
'''  
def file_read(fname):  
    txt = open(fname)  
    print(txt.read())  
file_read('text.txt')
```

**Output**



Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>> ===== RESTART =====  
>>>  
**What is Python language?**  
**Python is a widely used high-level, general-purpose, interpreted, dynamic programming language**  
**.**  
**Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java.**  
**Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles.**  
**It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.**  
**The best way we learn anything is by practice and exercise questions. We have started this section for those (beginner to intermediate) who are familiar with Python.**  
>>>

**B. Write a Python program to append text to a file and display the text.**

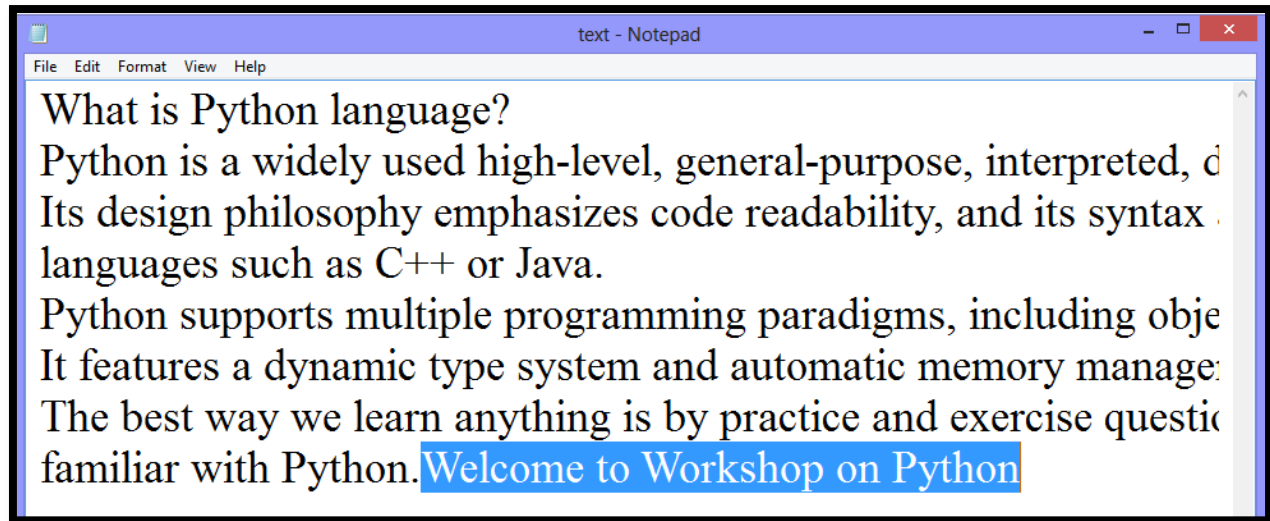
**Code:**

```
def main():  
    f=open("text.txt","a+")  
    f.write("Welcome to Workshop on Python")  
    f.close()  
if __name__=="__main__":  
    main()
```



```
6b1.py - C:/Users/Nitesh/Desktop/PYTHON WORKSHOP MATERIAL/SYIT PYTHON PRAX CODE/Prax6/6b1.py (2.7.13)  
File Edit Format Run Options Window Help  
def main():  
    f=open("text.txt","a+")  
    f.write("Welcome to Workshop on Python")  
    f.close()  
if __name__=="__main__":  
    main()
```

## Output:



**C. Write a Python program to read last n lines of a file.**

### Code:

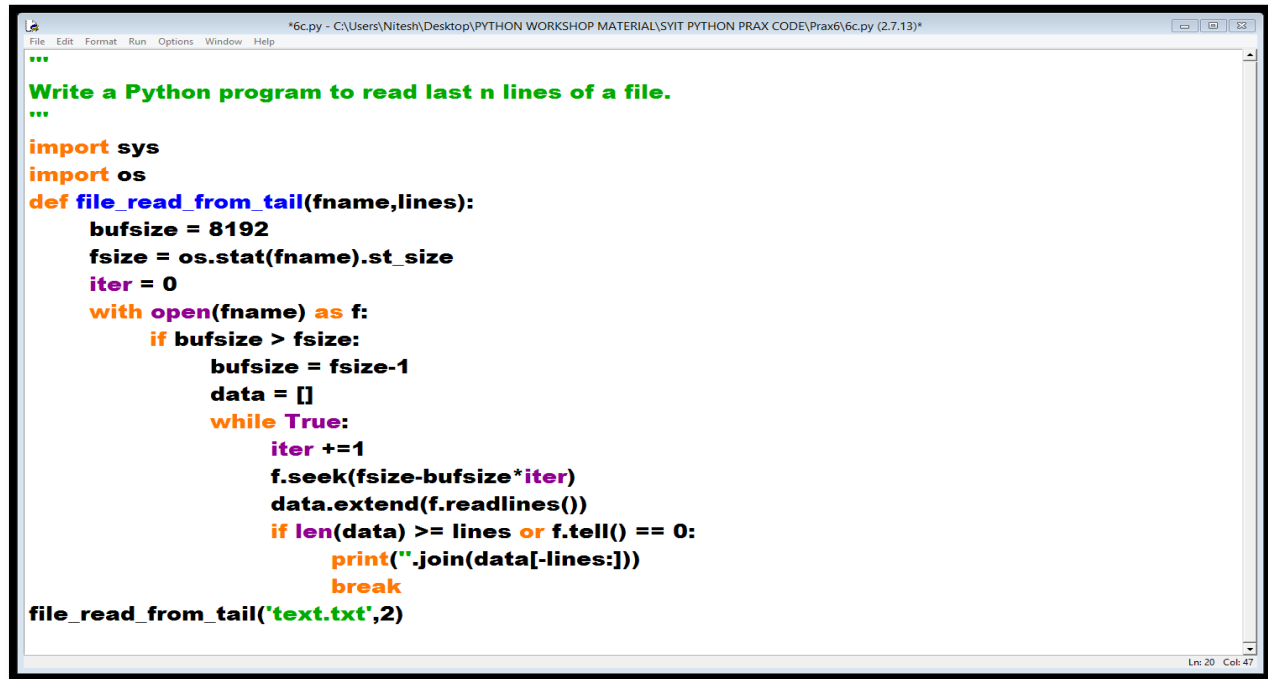
```
'''
```

**Write a Python program to read last n lines of a file.**

```
'''
```

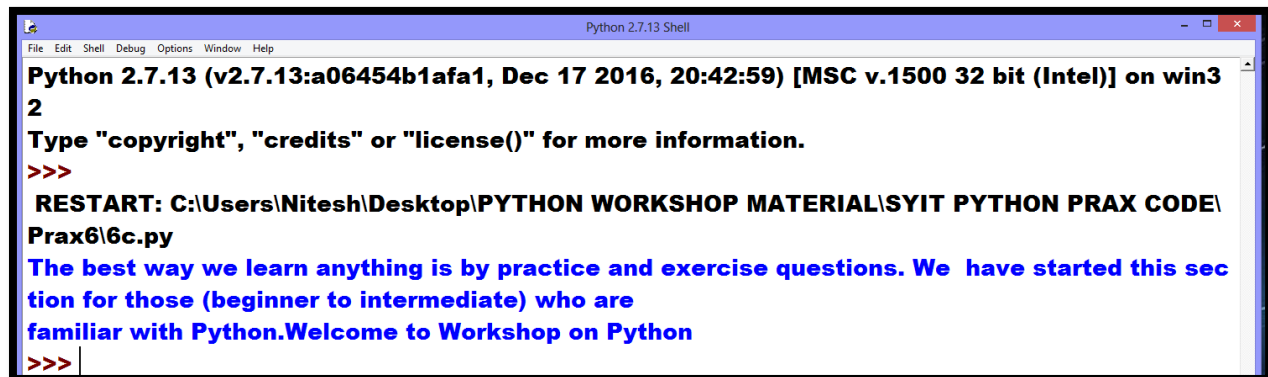
```
import sys
import os
def read_from_tail(fname,lines):
    bufsize = 8192
    fsize = os.stat(fname).st_size
    iter = 0
    with open(fname) as f:
        if bufsize>fsize:
            bufsize = fsize-1
            data = []
            while True:
                iter +=1
                f.seek(fsize-bufsize*iter)
                data.extend(f.readlines())
                if len(data) >= lines or f.tell() == 0:
```

```
        print('.join(data[-lines:]))
        break
file_read_from_tail('text.txt',2)
```



```
*6c.py - C:\Users\Nitesh\Desktop\PYTHON WORKSHOP MATERIAL\SYIT PYTHON PRAX CODE\Prax6\6c.py (2.7.13)*
File Edit Format Run Options Window Help
'''
Write a Python program to read last n lines of a file.
'''
import sys
import os
def file_read_from_tail(fname,lines):
    bufsize = 8192
    fsize = os.stat(fname).st_size
    iter = 0
    with open(fname) as f:
        if bufsize > fsize:
            bufsize = fsize-1
            data = []
            while True:
                iter +=1
                f.seek(fsize-bufsize*iter)
                data.extend(f.readlines())
                if len(data) >= lines or f.tell() == 0:
                    print('.join(data[-lines:]))
                    break
file_read_from_tail('text.txt',2)
```

## Output:



```
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\Nitesh\Desktop\PYTHON WORKSHOP MATERIAL\SYIT PYTHON PRAX CODE\Prax6\6c.py
The best way we learn anything is by practice and exercise questions. We have started this section for those (beginner to intermediate) who are familiar with Python.Welcome to Workshop on Python
>>>
```

## Practical No.-7

**Write the program for the following: ( class and objects)**

**A. Design a class that store the information of student and display the same**

```
class Student:
def __init__(self,name, sex,course,result):
    self.name=name
    self.sex=sex
self.course=course
self.result=result
    def display(self, name, sex, course, result):
        self.name=name
        self.sex=sex
self.course=course
self.result=result
        print ('Name:', name)
        print ('Sex:',sex)
        print ('course:',course)
        print ('result:', result)
```

### **Output:**

```
>>> s1 =Student()
>>> s1.display('AshwinMehta','M','B. Sc.(IT)','96.8%')
Name: Ashwin Mehta
Sex: M
course: B. Sc.(IT)
result: 96.8%
```

**B. Implement the concept of inheritance using python**

### **Code:**

```
class Shape:
    author= 'Ashwin Mehta'
def __init__(self,x,y):
    self.x=x
    self.y=y
```



```
def area(self,x,y):
self.x=x
self.y=y
    a=self.x*self.y
    print ('Area of a rectangle',a)
print (author)
class Square(Shape): #class Square inherits class Shape.
def __init__(self,x):
self.x=x
def area(self,x):
self.x=x
    a= self.x*self.x
    print('Area of a square',a)
```

### **Output:**

>>>

RESTART:

C:/Users/Welcome/AppData/Local/Programs/Python/Python36/inherit.py

Ashwin Mehta

>>> r=Shape()

>>>r.area(12,34)

Area of a rectangle 408

>>> s=Square()

>>>s.area(34)

Area of a square 1156

**C. Create a class called Numbers, which has a single class attribute called MULTIPLIER, and a constructor which takes the parameters x and y (these should all be numbers).**

**i. Write a method called add which returns the sum of the attributes x and y.**

**ii. Write a class method called multiply, which takes a single number parameter a and returns the product of a and MULTIPLIER.**

**iii. Write a static method called subtract, which takes two number parameters, b and c, and returns b - c. iv. Write a method called value which returns a tuple containing the values of x and y. Make this method into a property, and write a setter and a deleter for manipulating the values of x and y.**

```
class Numbers(object):
    def __init__(self,x,y):
        self.x=x
        self.y=y
    def add(self,x, y):
        self.x=x
        self.y=y
        return self.x+self.y
    def multiply(self,x):
        MULTIPLIER=7.4
        self.x=x
        return self.x*MULTIPLIER
```

**Output:**

```
>>> n3=Numbers()

>>> n3.multiply(4.78)

35.372000000000001

>>> n2=Numbers()

>>> n2.add(5.7,9.3)
```

## Practical No.-8

**8. Write the program for the following: (IDLE and exception handling)**

**A. Open a new file in IDLE (“New Window” in the “File” menu) and save it as geometry.py in the directory where you keep the files you create for this course. Then copy the functions you wrote for calculating volumes and areas in the “Control Flow and Functions” exercise into this file and save it. Now open a new file and save it in the same directory. You should now be able to import your own module like this: `import geometry` 16 Try and add `print dir(geometry)` to the file and run it. Now write a function `pointyShapeVolume(x, y, squareBase)` that calculates the volume of a square pyramid if `squareBase` is `True` and of a right circular cone if `squareBase` is `False`. `x` is the length of an edge on a square if `squareBase` is `True` and the radius of a circle when `squareBase` is `False`. `y` is the height of the object. First use `squareBase` to distinguish the cases. Use the `circleArea` and `squareArea` from the `geometry` module to calculate the base areas.**

### Code:-

```
import geometry
defpointyShapeVolume(x, h, square):
    if square:
        base = geometry.squareArea(x)
    else:
        base = geometry.circleArea(x)
    return h * base / 3.0
print dir(geometry)
print pointyShapeVolume(4, 2.6, True)
print pointyShapeVolume(4, 2.6, False)
```

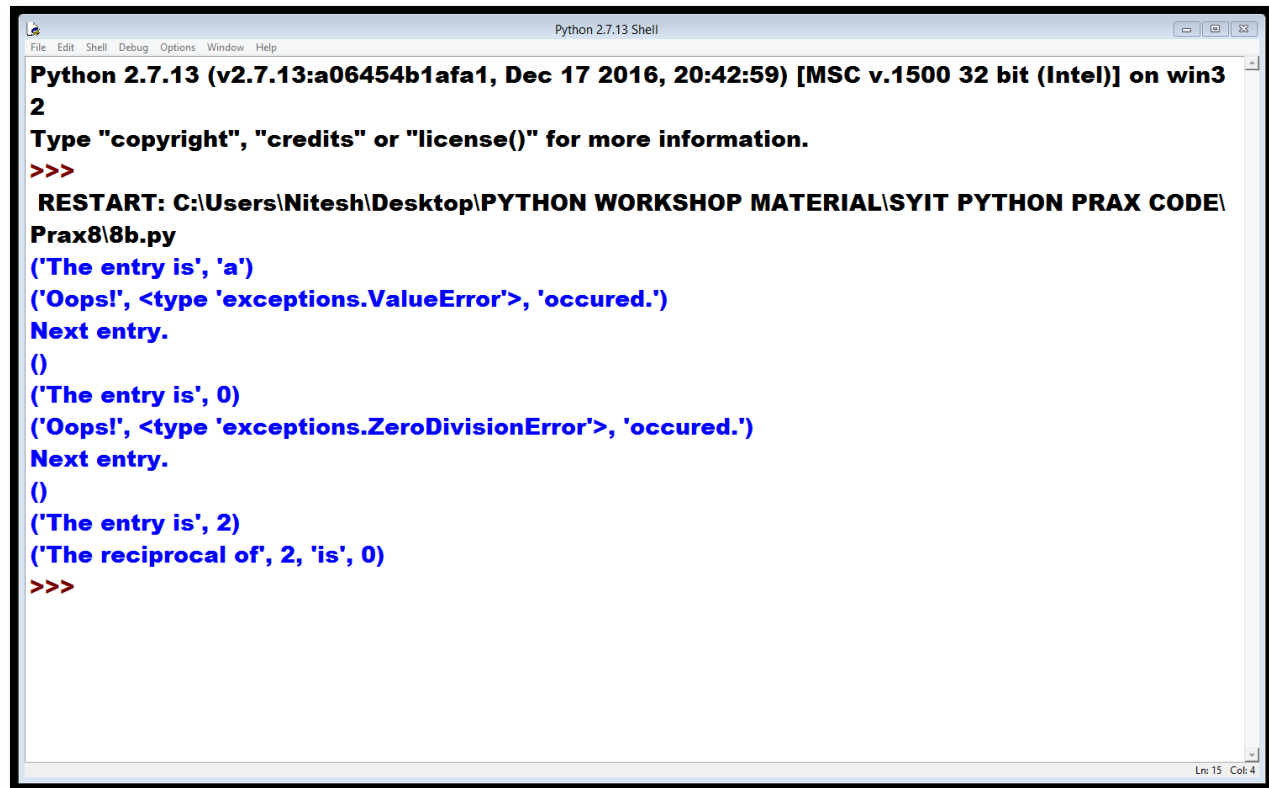
```
8a.py - C:\Users\Nitesh\Desktop\PYTHON WORKSHOP MATERIAL\SYIT PYTHON PRAX CODE\Prax8\8a.py (2.7.13)
File Edit Format Run Options Window Help
import geometry
def pointyShapeVolume(x, h, square):
    if square:
        base = geometry.squareArea(x)
    else:
        base = geometry.circleArea(x)
    return h * base / 3.0
print dir(geometry)
print pointyShapeVolume(4, 2.6, True)
print pointyShapeVolume(4, 2.6, False)
```

B. Write a program to implement exception handling.

```
import sys
randomList = ['a', 0, 2]
for entry in randomList:
    try:
        print("The entry is", entry)
        r = 1/int(entry)
        break
    except:
        print("Oops!",sys.exc_info()[0],"occured.")
        print("Next entry.")
        print()
print("The reciprocal of",entry,"is",r)
```

```
8b.py - C:\Users\Nitesh\Desktop\PYTHON WORKSHOP MATERIAL\SYIT PYTHON PRAX CODE\Prax8\8b.py (2.7.13)
File Edit Format Run Options Window Help
...
Write a program to implement exception handling.
...
import sys
randomList = ['a', 0, 2]
for entry in randomList:
    try:
        print("The entry is", entry)
        r = 1/int(entry)
        break
    except:
        print("Oops!",sys.exc_info()[0],"occured.")
        print("Next entry.")
        print()
print("The reciprocal of",entry,"is",r)
```

Output:



```
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\Nitesh\Desktop\PYTHON WORKSHOP MATERIAL\SYIT PYTHON PRAX CODE\
Prax8\8b.py
('The entry is', 'a')
('Oops!', <type 'exceptions.ValueError'>, 'occured.')
Next entry.
()
('The entry is', 0)
('Oops!', <type 'exceptions.ZeroDivisionError'>, 'occured.')
Next entry.
()
('The entry is', 2)
('The reciprocal of', 2, 'is', 0)
>>>
```

Ln: 15 Col: 4

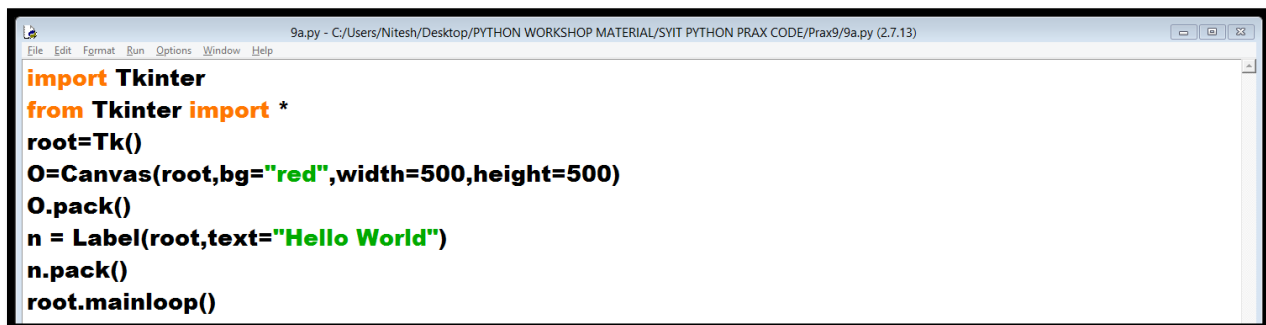
## Practical No.9

**Write the program for the following: (Widget - GUI)**

**A. Try to configure the widget with various options like: bg="red", family="times", size=18**

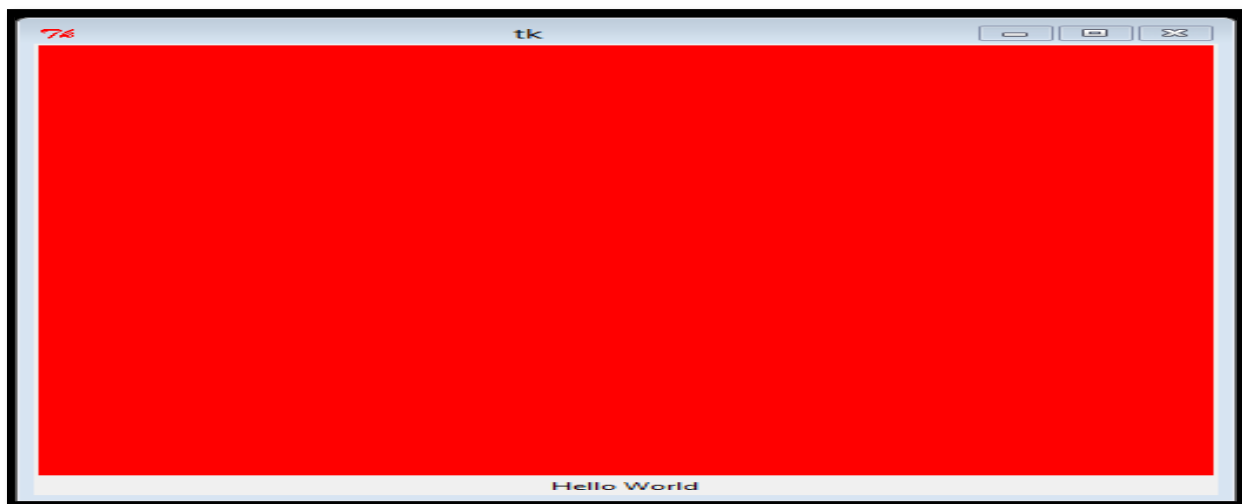
**Code:**

```
import Tkinter
from Tkinter import *
root=Tk()
O=Canvas(root,bg="red",width=500,height=500)
O.pack()
n = Label(root,text="Hello World")
n.pack()
root.mainloop()
```

A screenshot of a Python IDE window titled '9a.py - C:/Users/Nitesh/Desktop/PYTHON WORKSHOP MATERIAL/SYT PYTHON PRAX CODE/Prax9/9a.py (2.7.13)'. The code editor shows the following Python code: 

```
import Tkinter
from Tkinter import *
root=Tk()
O=Canvas(root,bg="red",width=500,height=500)
O.pack()
n = Label(root,text="Hello World")
n.pack()
root.mainloop()
```

**OutPut:-**

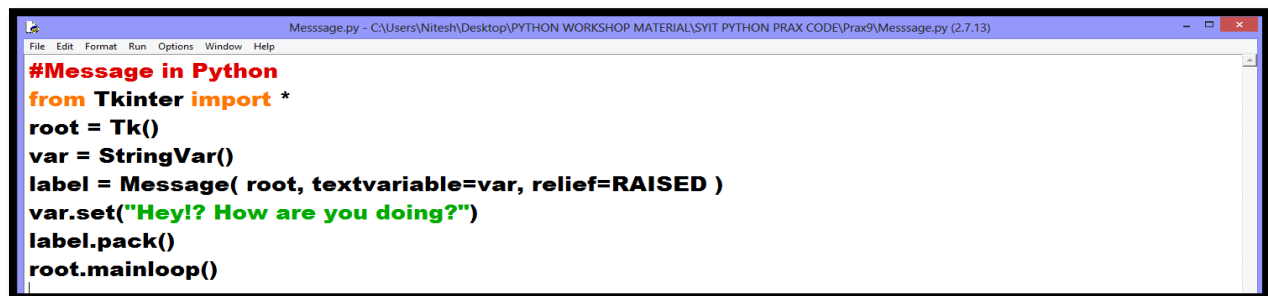


**B. Try to change the widget type and configuration options to experiment with other widget types like Message, Button, Entry, Checkbutton, Radiobutton, Scale etc.**

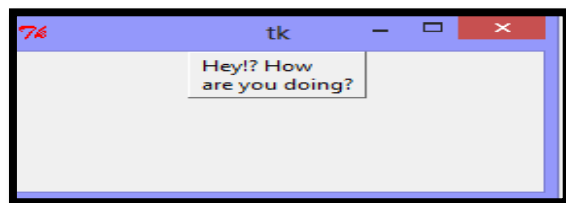
**Code:**

**Message.py**

```
#Message in Python
from Tkinter import *
root = Tk()
var = StringVar()
label = Message( root, textvariable=var, relief=RAISED )
var.set("Hey!? How are you doing?")
label.pack()
root.mainloop()
```



**Output:-**




**Button.py**

**Code:-**

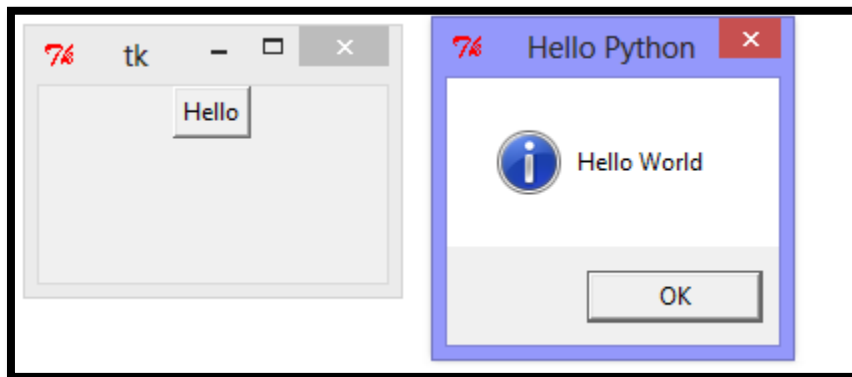
```
#Button in Python
import Tkinter
import tkMessageBox
top = Tkinter.Tk()
```

```
def helloCallBack():  
    tkMessageBox.showinfo( "Hello Python", "Hello World")  
B = Tkinter.Button(top, text ="Hello", command = helloCallBack)  
B.pack()  
top.mainloop()
```



```
Button.py - C:\Users\Nitesh\Desktop\PYTHON WORKSHOP MATERIAL\SYIT PYTHON PRAX CODE\Prax9\Button.py (2.7.13)  
File Edit Format Run Options Window Help  
#Button in Python  
import Tkinter  
import tkMessageBox  
top = Tkinter.Tk()  
def helloCallBack():  
    tkMessageBox.showinfo( "Hello Python", "Hello World")  
B = Tkinter.Button(top, text ="Hello", command = helloCallBack)  
B.pack()  
top.mainloop()
```

## Output:



## Entry.py

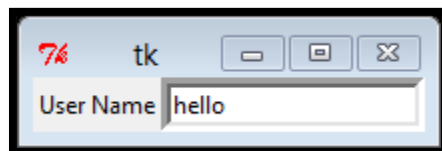
### Code:

```
#Entry in Python  
from Tkinter import *  
top = Tk()  
L1 = Label(top, text="User Name")  
L1.pack( side = LEFT)  
E1 = Entry(top, bd =5)  
E1.pack(side = RIGHT)  
top.mainloop()
```



```
Entry.py - C:\Users\Nitesh\Desktop\PYTHON WORKSHOP MATERIAL\SYIT PYTHON PRAX CODE\Prax9\Entry.py (2.7.13)
File Edit Format Run Options Window Help
#Entry in Python
from Tkinter import *
top = Tk()
L1 = Label(top, text="User Name")
L1.pack( side = LEFT)
E1 = Entry(top, bd =5)
E1.pack(side = RIGHT)
top.mainloop()
```

**Output:-**



### **CheckBoxButton.py**

#CheckBox In Python

```
from Tkinter import *
```

```
import tkMessageBox
```

```
import Tkinter
```

```
top = Tkinter.Tk()
```

```
CheckVar1 = IntVar()
```

```
CheckVar2 = IntVar()
```

```
C1 = Checkbutton(top, text = "Music", variable = CheckVar1, \
onvalue = 1, offvalue = 0, height=5, \
width = 20)
```

```
C2 = Checkbutton(top, text = "Video", variable = CheckVar2, \
onvalue = 1, offvalue = 0, height=5, \
width = 20)
```

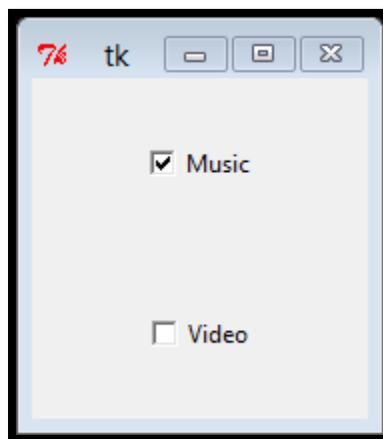
```
C1.pack()
```

```
C2.pack()
```

```
top.mainloop()
```

```
*CheckBoxButton.py - C:\Users\Nitesh\Desktop\PYTHON WORKSHOP MATERIAL\SVIT PYTHON PRAX CODE\Prax9\CheckBoxButton.py (2.7.13)*
File Edit Format Run Options Window Help
#CheckBoxButton In Python
from Tkinter import *
import tkMessageBox
import Tkinter
top = Tkinter.Tk()
CheckVar1 = IntVar()
CheckVar2 = IntVar()
C1 = Checkbutton(top, text = "Music", variable = CheckVar1, \
                 onvalue = 1, offvalue = 0, height=5, \
                 width = 20)
C2 = Checkbutton(top, text = "Video", variable = CheckVar2, \
                 onvalue = 1, offvalue = 0, height=5, \
                 width = 20)
C1.pack()
C2.pack()
top.mainloop()
```

**Output:-**



## **RadioButton.py**

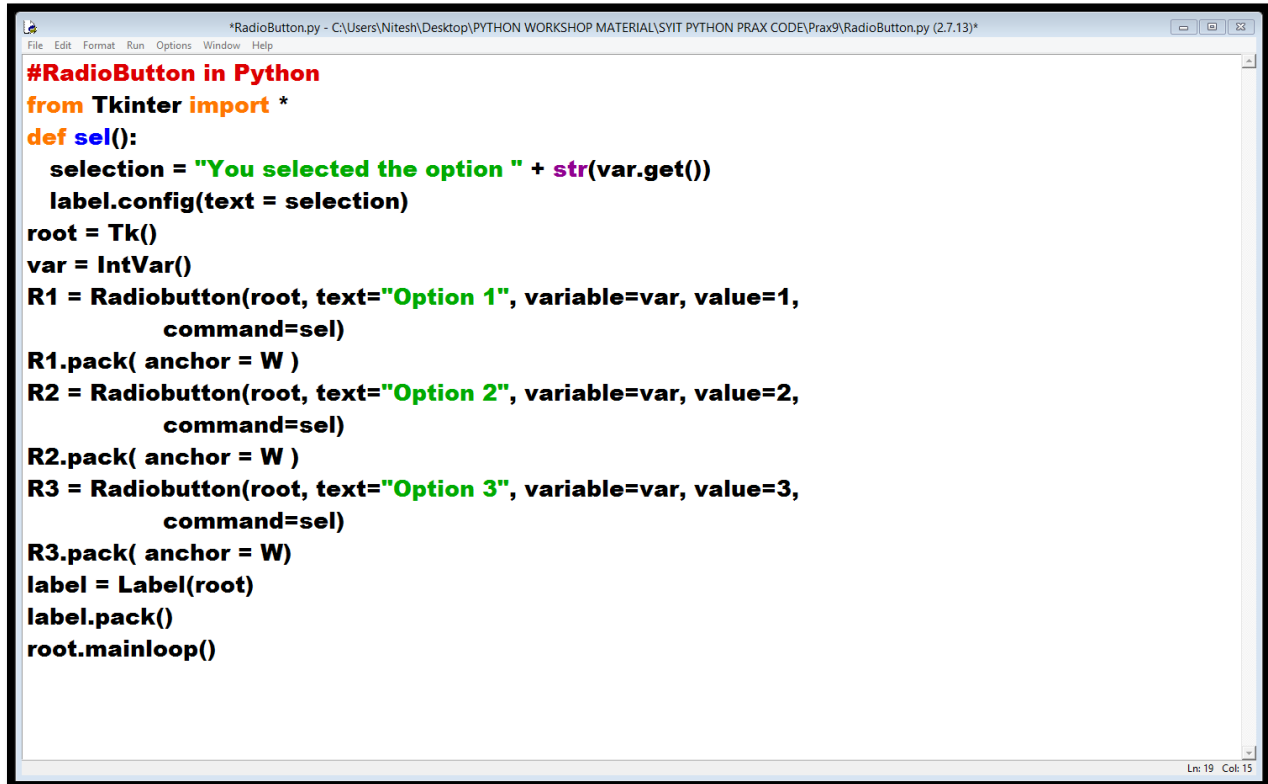
### **Code:**

```
#RadioButton in Python
from Tkinter import *
defsel():
    selection = "You selected the option " + str(var.get())
    label.config(text = selection)
root = Tk()
var = IntVar()
R1 = Radiobutton(root, text="Option 1", variable=var, value=1,
                 command=sel)
R1.pack( anchor = W )
R2 = Radiobutton(root, text="Option 2", variable=var, value=2,
```

```

        command=sel)
R2.pack( anchor = W )
R3 = Radiobutton(root, text="Option 3", variable=var, value=3,
        command=sel)
R3.pack( anchor = W)
label = Label(root)
label.pack()
root.mainloop()

```

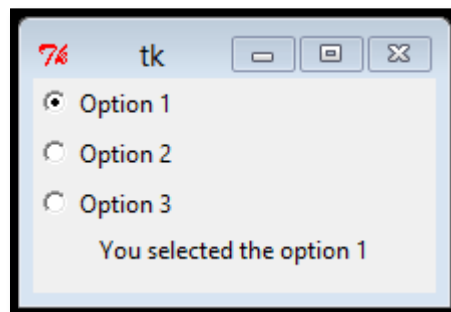


```

*RadioButton.py - C:\Users\Nitesh\Desktop\PYTHON WORKSHOP MATERIAL\SVIT PYTHON PRAX CODE\Prax9\RadioButton.py (2.7.13)*
File Edit Format Run Options Window Help
#RadioButton in Python
from Tkinter import *
def sel():
    selection = "You selected the option " + str(var.get())
    label.config(text = selection)
root = Tk()
var = IntVar()
R1 = Radiobutton(root, text="Option 1", variable=var, value=1,
    command=sel)
R1.pack( anchor = W )
R2 = Radiobutton(root, text="Option 2", variable=var, value=2,
    command=sel)
R2.pack( anchor = W )
R3 = Radiobutton(root, text="Option 3", variable=var, value=3,
    command=sel)
R3.pack( anchor = W)
label = Label(root)
label.pack()
root.mainloop()
Ln: 19 Col: 15

```

**Output:**



**Scale.py**

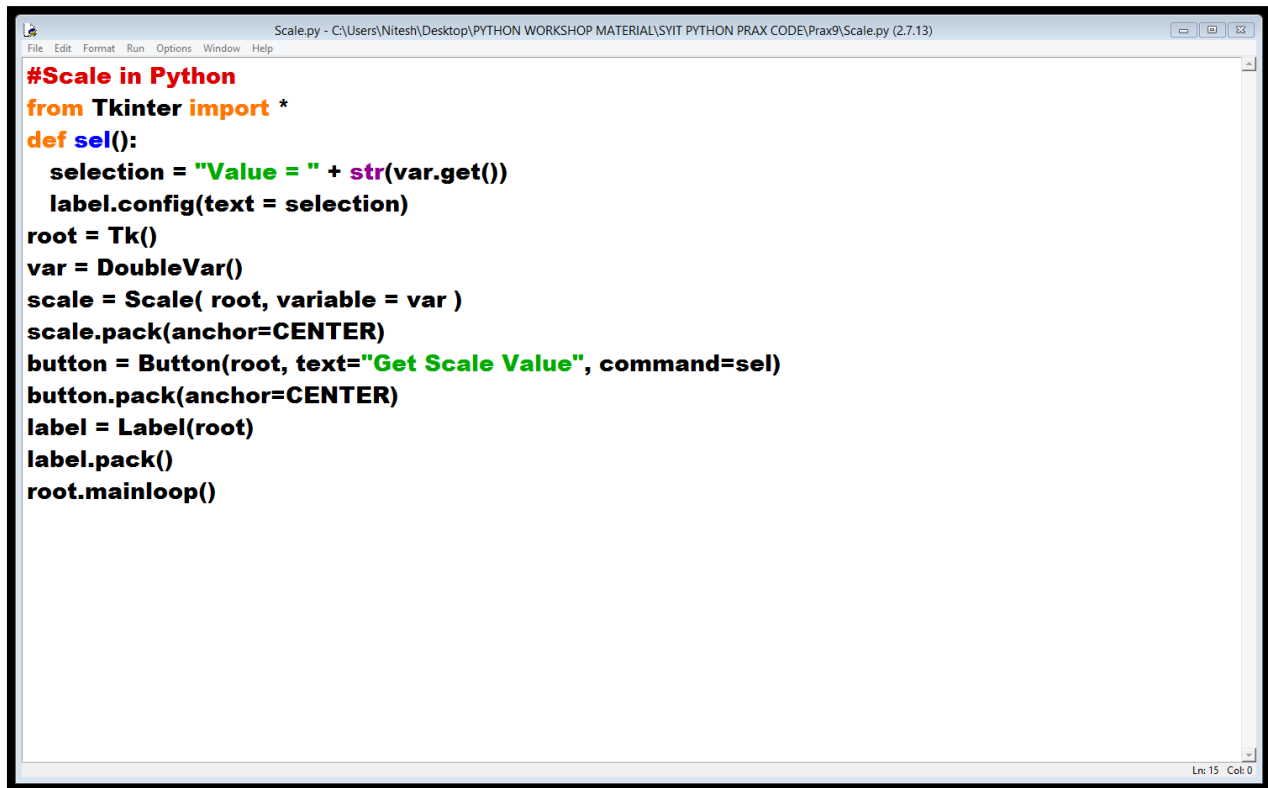
**Code:-**

```

#Scale in Python
from Tkinter import *

```

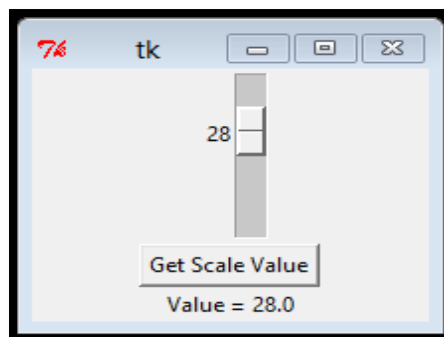
```
defsel():  
    selection = "Value = " + str(var.get())  
    label.config(text = selection)  
root = Tk()  
var = DoubleVar()  
scale = Scale( root, variable = var )  
scale.pack(anchor=CENTER)  
button = Button(root, text="Get Scale Value", command=sel)  
button.pack(anchor=CENTER)  
label = Label(root)  
label.pack()  
root.mainloop()
```



The screenshot shows a Python IDE window titled "Scale.py - C:\Users\Nitesh\Desktop\PYTHON WORKSHOP MATERIAL\SYIT PYTHON PRAX CODE\Prax9\Scale.py (2.7.13)". The code is as follows:

```
#Scale in Python  
from Tkinter import *  
def sel():  
    selection = "Value = " + str(var.get())  
    label.config(text = selection)  
root = Tk()  
var = DoubleVar()  
scale = Scale( root, variable = var )  
scale.pack(anchor=CENTER)  
button = Button(root, text="Get Scale Value", command=sel)  
button.pack(anchor=CENTER)  
label = Label(root)  
label.pack()  
root.mainloop()
```

**Output:-**

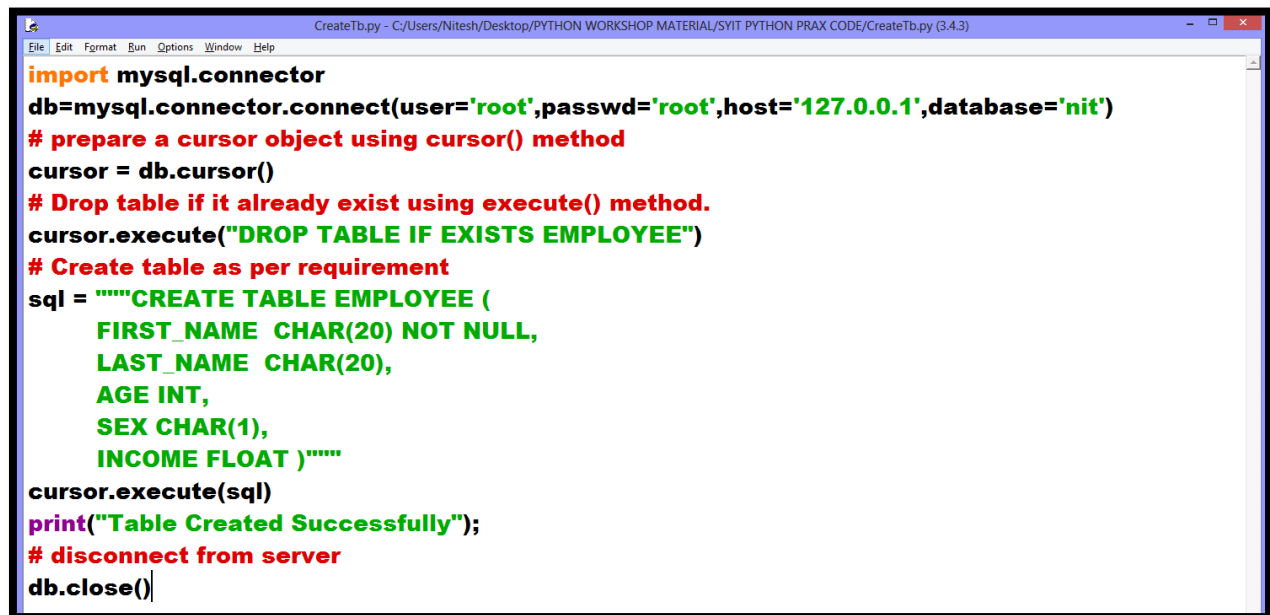


## Practical No.10

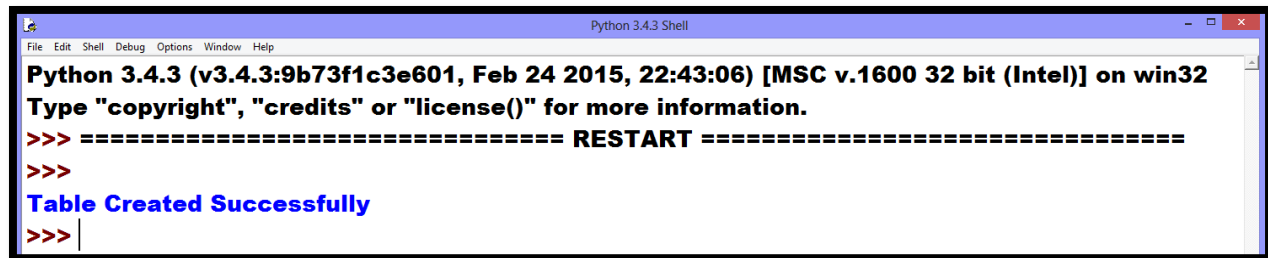
**Design the database applications for the following: (Refer database Chapter)**

**A. Design a simple database application that stores the records and retrieve the same.**

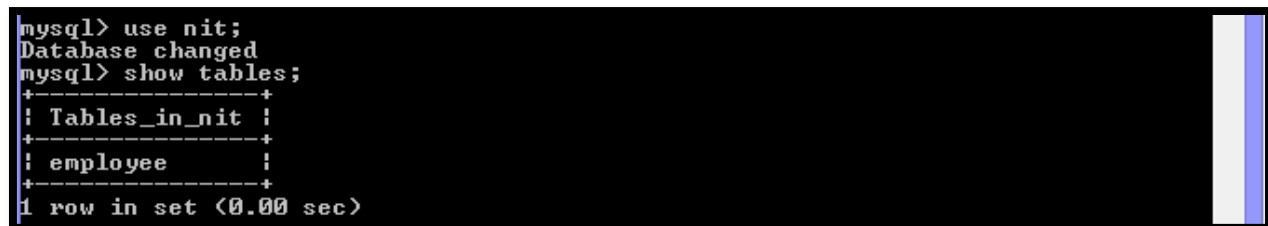
```
import mysql.connector
db=mysql.connector.connect(user='root',passwd='root',host='127.0.0.1',database='nit')
# prepare a cursor object using cursor() method
cursor = db.cursor()
# Drop table if it already exist using execute() method.
cursor.execute("DROP TABLE IF EXISTS EMPLOYEE")
# Create table as per requirement
sql = """CREATE TABLE EMPLOYEE (
        FIRST_NAME CHAR(20) NOT NULL,
        LAST_NAME CHAR(20),
        AGE INT,
        SEX CHAR(1),
        INCOME FLOAT )"""
cursor.execute(sql)
print("Table Created Successfully");
# disconnect from server
db.close()
```

A screenshot of a Python IDE window titled 'CreateTb.py - C:/Users/Nitesh/Desktop/PYTHON WORKSHOP MATERIAL/SYIT PYTHON PRAX CODE/CreateTb.py (3.4.3)'. The code is displayed with syntax highlighting: keywords in blue, strings in red, and comments in green. The code is identical to the one shown in the text block above, demonstrating the steps to connect to a MySQL database, drop an existing table, create a new table with specific columns, and disconnect from the server.

```
import mysql.connector
db=mysql.connector.connect(user='root',passwd='root',host='127.0.0.1',database='nit')
# prepare a cursor object using cursor() method
cursor = db.cursor()
# Drop table if it already exist using execute() method.
cursor.execute("DROP TABLE IF EXISTS EMPLOYEE")
# Create table as per requirement
sql = """CREATE TABLE EMPLOYEE (
        FIRST_NAME CHAR(20) NOT NULL,
        LAST_NAME CHAR(20),
        AGE INT,
        SEX CHAR(1),
        INCOME FLOAT )"""
cursor.execute(sql)
print("Table Created Successfully");
# disconnect from server
db.close()
```



```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Table Created Successfully
>>>
```



```
mysql> use nit;
Database changed
mysql> show tables;
+-----+
| Tables_in_nit |
+-----+
| employee      |
+-----+
1 row in set (0.00 sec)
```

```
import mysql.connector
db=mysql.connector.connect(user='root',passwd='root',host='127.0.0.1',database='python_mysql')
# prepare a cursor object using cursor() method
cursor = db.cursor()
# Prepare SQL query to INSERT a record into the database.
sql = """INSERT INTO EMPLOYEE(FIRST_NAME,
            LAST_NAME, AGE, SEX, INCOME)
            VALUES ('Nitesh', 'Shukla', 23, 'M', 20000)"""
try:
    # Execute the SQL command
    cursor.execute(sql)
    print ("Data Inserted Successfully...!")
    # Commit your changes in the database
    db.commit()
except:
    # Rollback in case there is any error
    db.rollback()
# disconnect from server
db.close()
```

```
mytest.py - C:/Users/Nitesh/Desktop/mytest.py (3.4.3)
File Edit Format Run Options Window Help
import mysql.connector
db=mysql.connector.connect(user='root',passwd='root',host='127.0.0.1',database='nit')
# prepare a cursor object using cursor() method
cursor = db.cursor()
# Prepare SQL query to INSERT a record into the database.
sql = """INSERT INTO EMPLOYEE(FIRST_NAME,
    LAST_NAME, AGE, SEX, INCOME)
    VALUES ('Nitesh', 'Shukla', 20, 'M', 20000)"""
try:
    # Execute the SQL command
    cursor.execute(sql)
    print("Data Inserted Successfully..")
    # Commit your changes in the database
    db.commit()
except:
    # Rollback in case there is any error
    db.rollback()
# disconnect from server
db.close()
```

```
mysql> select * from employee;
+-----+-----+-----+-----+-----+
| FIRST_NAME | LAST_NAME | AGE | SEX | INCOME |
+-----+-----+-----+-----+-----+
| Nitesh     | Shukla    | 20  | M   | 20000   |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

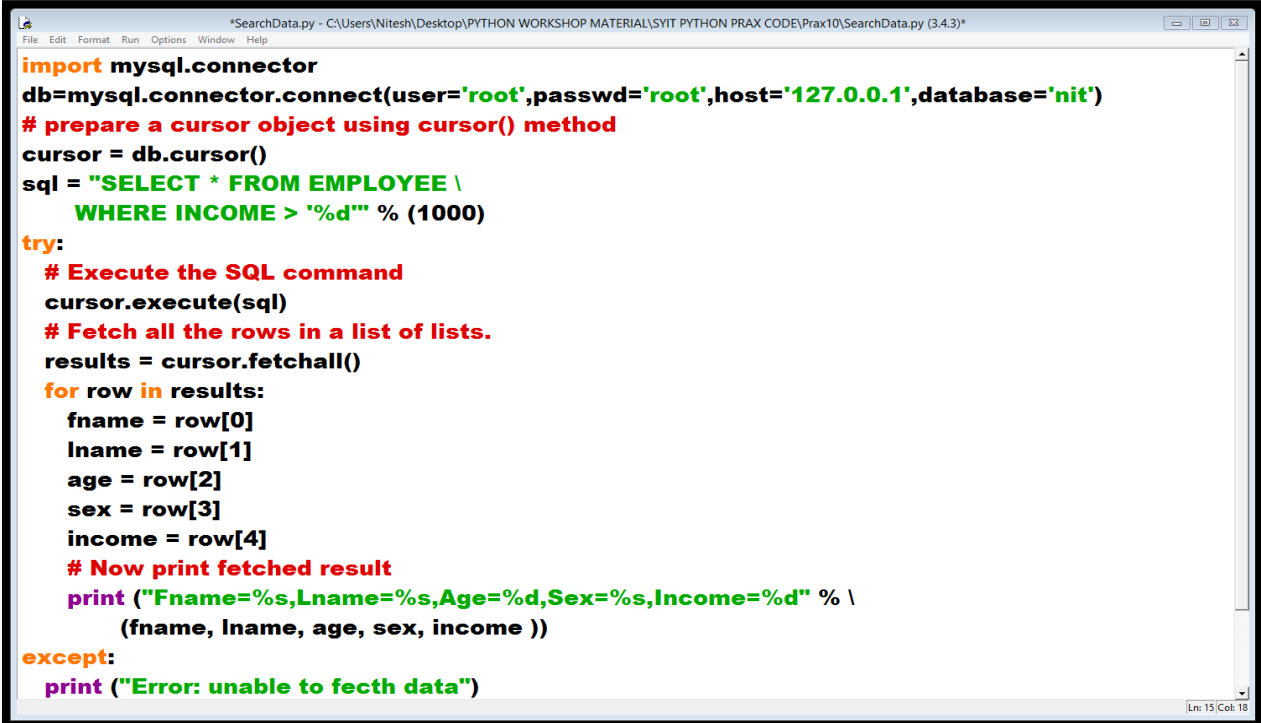
**B. Design a database application to search the specified record from the database.**

```
import mysql.connector
db=mysql.connector.connect(user='root',passwd='root',host='127.0.0.1',database='n
it')
# prepare a cursor object using cursor() method
cursor = db.cursor()
sql = "SELECT * FROM EMPLOYEE \
    WHERE INCOME > '%d'" % (1000)
try:
    # Execute the SQL command
    cursor.execute(sql)
    # Fetch all the rows in a list of lists.
    results = cursor.fetchall()
    for row in results:
```

```

fname = row[0]
lname = row[1]
    age = row[2]
    sex = row[3]
    income = row[4]
    # Now print fetched result
    print ("Fname=%s,Lname=%s,Age=%d,Sex=%s,Income=%d" % \
        (fname, lname, age, sex, income ))
except:
    print ("Error: unable to fetch data")
# disconnect from server
db.close()

```



```

*SearchData.py - C:\Users\Nitesh\Desktop\PYTHON WORKSHOP MATERIAL\SYIT PYTHON PRAX CODE\Prax10\SearchData.py (3.4.3)*
File Edit Format Run Options Window Help
import mysql.connector
db=mysql.connector.connect(user='root',passwd='root',host='127.0.0.1',database='nit')
# prepare a cursor object using cursor() method
cursor = db.cursor()
sql = "SELECT * FROM EMPLOYEE \
      WHERE INCOME > '%d'" % (1000)
try:
    # Execute the SQL command
    cursor.execute(sql)
    # Fetch all the rows in a list of lists.
    results = cursor.fetchall()
    for row in results:
        fname = row[0]
        lname = row[1]
        age = row[2]
        sex = row[3]
        income = row[4]
        # Now print fetched result
        print ("Fname=%s,Lname=%s,Age=%d,Sex=%s,Income=%d" % \
            (fname, lname, age, sex, income ))
except:
    print ("Error: unable to fetch data")

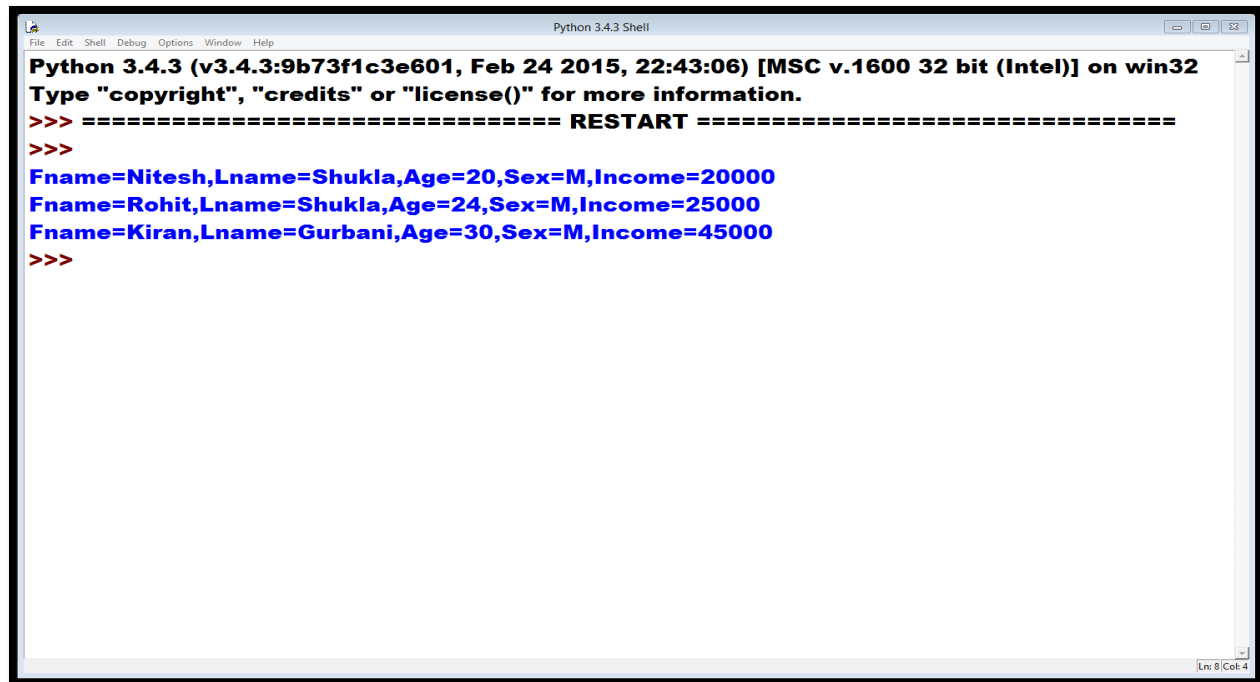
```

```

# disconnect from server
db.close()

```





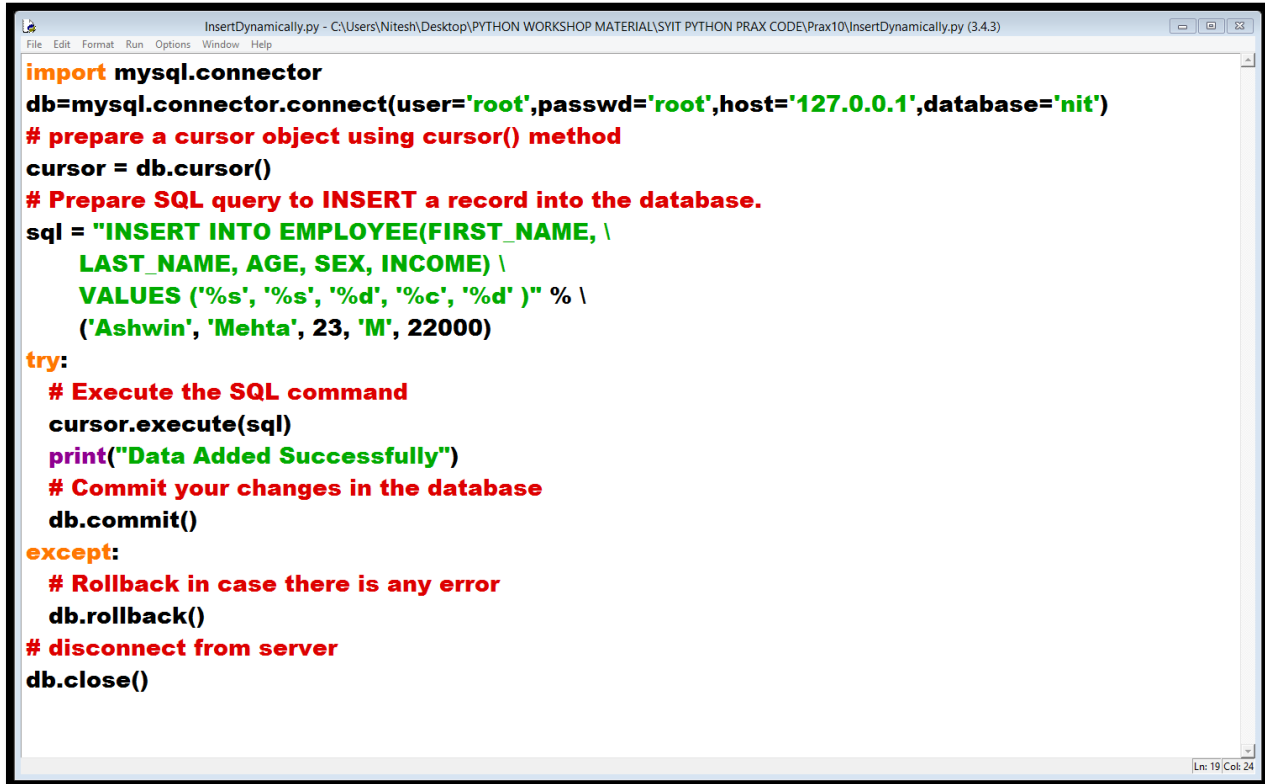
```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Fname=Nitesh,Lname=Shukla,Age=20,Sex=M,Income=20000
Fname=Rohit,Lname=Shukla,Age=24,Sex=M,Income=25000
Fname=Kiran,Lname=Gurbani,Age=30,Sex=M,Income=45000
>>>
```

**C. Design a database application to that allows the user to add, delete and modify the records.**

DataAdd.py

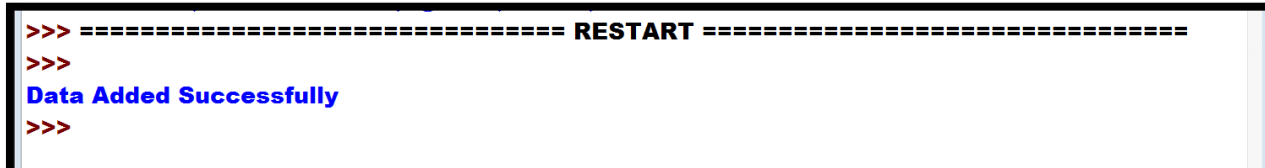
```
import mysql.connector
db=mysql.connector.connect(user='root',passwd='root',host='127.0.0.1',database='n
it')
# prepare a cursor object using cursor() method
cursor = db.cursor()
# Prepare SQL query to INSERT a record into the database.
sql = "INSERT INTO EMPLOYEE(FIRST_NAME, \
      LAST_NAME, AGE, SEX, INCOME) \
      VALUES ('%s', '%s', '%d', '%c', '%d' )" % \
      ('Ashwin', 'Mehta', 23, 'M', 22000)
try:
    # Execute the SQL command
    cursor.execute(sql)
    print("Data Added Successfully")
    # Commit your changes in the database
    db.commit()
except:
    # Rollback in case there is any error
```

```
db.rollback()
# disconnect from server
db.close()
```

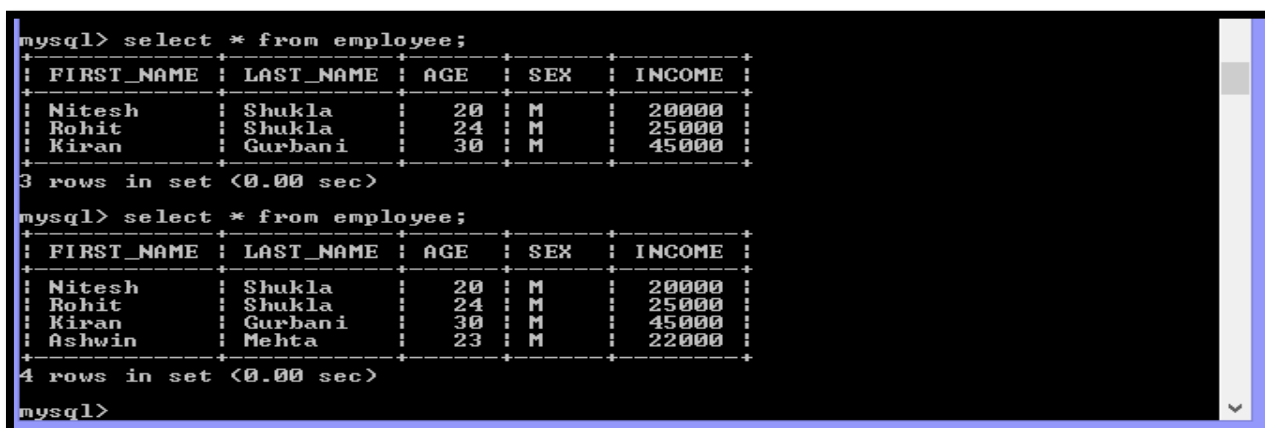


```
InsertDynamically.py - C:\Users\Nitesh\Desktop\PYTHON WORKSHOP MATERIAL\SVIT PYTHON PRAX CODE\Prax10\InsertDynamically.py (3.4.3)
File Edit Format Run Options Window Help

import mysql.connector
db=mysql.connector.connect(user='root',passwd='root',host='127.0.0.1',database='nit')
# prepare a cursor object using cursor() method
cursor = db.cursor()
# Prepare SQL query to INSERT a record into the database.
sql = "INSERT INTO EMPLOYEE(FIRST_NAME, \
      LAST_NAME, AGE, SEX, INCOME) \
      VALUES ('%s', '%s', '%d', '%c', '%d' )" % \
      ('Ashwin', 'Mehta', 23, 'M', 22000)
try:
    # Execute the SQL command
    cursor.execute(sql)
    print("Data Added Successfully")
    # Commit your changes in the database
    db.commit()
except:
    # Rollback in case there is any error
    db.rollback()
# disconnect from server
db.close()
```



```
>>> ===== RESTART =====
>>>
Data Added Successfully
>>>
```



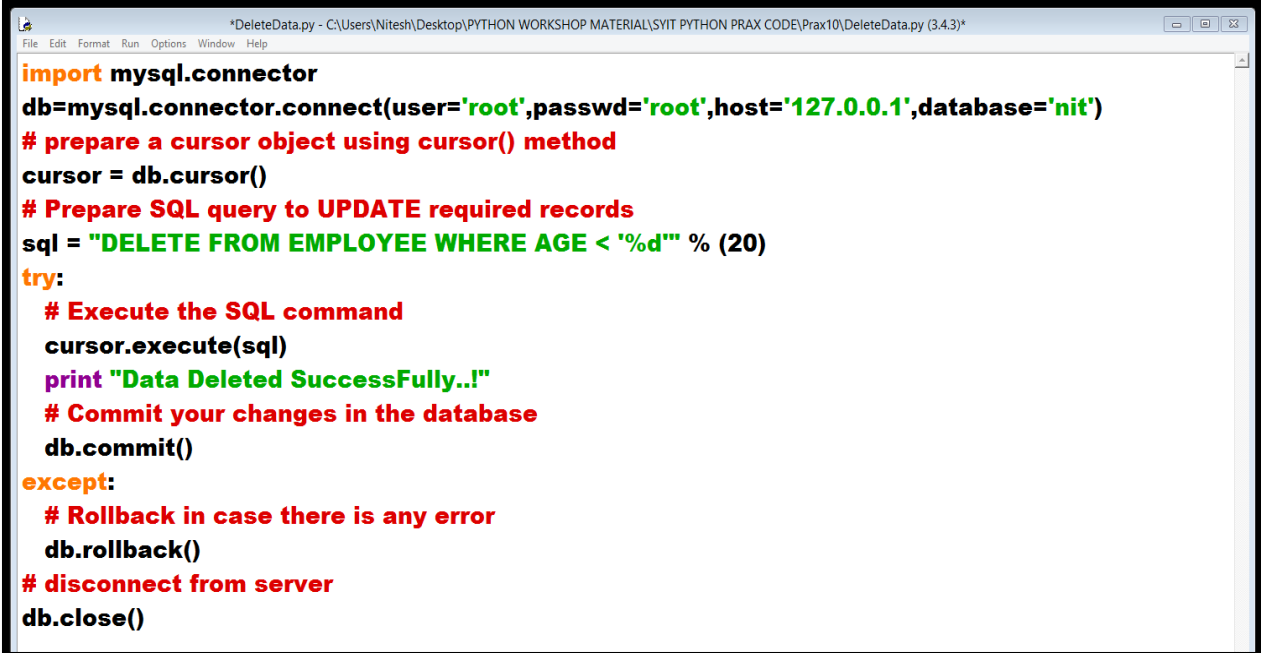
```
mysql> select * from employee;
+----+-----+-----+-----+-----+
| FIRST_NAME | LAST_NAME | AGE | SEX | INCOME |
+----+-----+-----+-----+-----+
| Nitesh     | Shukla   | 20  | M   | 20000  |
| Rohit      | Shukla   | 24  | M   | 25000  |
| Kiran      | Gurbani  | 30  | M   | 45000  |
+----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from employee;
+----+-----+-----+-----+-----+
| FIRST_NAME | LAST_NAME | AGE | SEX | INCOME |
+----+-----+-----+-----+-----+
| Nitesh     | Shukla   | 20  | M   | 20000  |
| Rohit      | Shukla   | 24  | M   | 25000  |
| Kiran      | Gurbani  | 30  | M   | 45000  |
| Ashwin     | Mehta    | 23  | M   | 22000  |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

## Delete.py

```
import mysql.connector
db=mysql.connector.connect(user='root',passwd='root',host='127.0.0.1',database='nit')
# prepare a cursor object using cursor() method
cursor = db.cursor()
# Prepare SQL query to UPDATE required records
sql = "DELETE FROM EMPLOYEE WHERE AGE < '%d'" % (20)
try:
    # Execute the SQL command
    cursor.execute(sql)
    print "Data Deleted Successfully..!"
    # Commit your changes in the database
    db.commit()
except:
    # Rollback in case there is any error
    db.rollback()
# disconnect from server
db.close()
```

A screenshot of a Python IDE window titled "DeleteData.py - C:\Users\Nitesh\Desktop\PYTHON WORKSHOP MATERIAL\SYIT PYTHON PRAX CODE\Prax10\DeleteData.py (3.4.3)". The window shows the same Python code as the previous block, with syntax highlighting: keywords in orange, strings in green, and comments in red. The code is as follows:

```
import mysql.connector
db=mysql.connector.connect(user='root',passwd='root',host='127.0.0.1',database='nit')
# prepare a cursor object using cursor() method
cursor = db.cursor()
# Prepare SQL query to UPDATE required records
sql = "DELETE FROM EMPLOYEE WHERE AGE < '%d'" % (20)
try:
    # Execute the SQL command
    cursor.execute(sql)
    print "Data Deleted Successfully..!"
    # Commit your changes in the database
    db.commit()
except:
    # Rollback in case there is any error
    db.rollback()
# disconnect from server
db.close()
```

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Data Deleted Successfully..!
>>>
```

```
mysql> select * from employee;
+-----+-----+-----+-----+-----+
| FIRST_NAME | LAST_NAME | AGE | SEX | INCOME |
+-----+-----+-----+-----+-----+
| Nitesh     | Shukla    | 20  | M   | 20000   |
| Rohit      | Shukla    | 24  | M   | 25000   |
| Kiran      | Gurbani   | 30  | M   | 45000   |
| Ashwin     | Mehta     | 23  | M   | 22000   |
| brij       | Shukla    | 18  | M   | 25000   |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from employee;
+-----+-----+-----+-----+-----+
| FIRST_NAME | LAST_NAME | AGE | SEX | INCOME |
+-----+-----+-----+-----+-----+
| Nitesh     | Shukla    | 20  | M   | 20000   |
| Rohit      | Shukla    | 24  | M   | 25000   |
| Kiran      | Gurbani   | 30  | M   | 45000   |
| Ashwin     | Mehta     | 23  | M   | 22000   |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

## Update.py

```
import mysql.connector
db=mysql.connector.connect(user='root',passwd='root',host='127.0.0.1',database='n
it')
# prepare a cursor object using cursor() method
cursor = db.cursor()
# Prepare SQL query to UPDATE required records
sql = "DELETE FROM EMPLOYEE WHERE AGE < '%d'" % (20)
try:
    # Execute the SQL command
    cursor.execute(sql)
    print ("Data Deleted Successfully..!")
    # Commit your changes in the database
    db.commit()
except:
    # Rollback in case there is any error
    db.rollback()
# disconnect from server
```

db.close()

```
Update.py - C:\Users\Nitesh\Desktop\PYTHON WORKSHOP MATERIAL\SVIT PYTHON PRAX CODE\Prax10\Update.py (3.4.3)
File Edit Format Run Options Window Help
import mysql.connector
db=mysql.connector.connect(user='root',passwd='root',host='127.0.0.1',database='nit')
# prepare a cursor object using cursor() method
cursor = db.cursor()
# Prepare SQL query to UPDATE required records
sql = "UPDATE EMPLOYEE SET AGE = AGE + 1 \
      WHERE SEX = '%c'" % ('M')
try:
    # Execute the SQL command
    cursor.execute(sql)
    # Commit your changes in the database
    db.commit()
except:
    # Rollback in case there is any error
    db.rollback()
# disconnect from server
db.close()
```

```
mysql> select * from employee;
+-----+-----+-----+-----+-----+
| FIRST_NAME | LAST_NAME | AGE | SEX | INCOME |
+-----+-----+-----+-----+-----+
| Nitesh     | Shukla   | 20  | M   | 20000  |
| Rohit      | Shukla   | 24  | M   | 25000  |
| Kiran      | Gurbani  | 30  | M   | 45000  |
| Ashwin     | Mehta    | 23  | M   | 22000  |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from employee;
+-----+-----+-----+-----+-----+
| FIRST_NAME | LAST_NAME | AGE | SEX | INCOME |
+-----+-----+-----+-----+-----+
| Nitesh     | Shukla   | 21  | M   | 20000  |
| Rohit      | Shukla   | 25  | M   | 25000  |
| Kiran      | Gurbani  | 31  | M   | 45000  |
| Ashwin     | Mehta    | 24  | M   | 22000  |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```