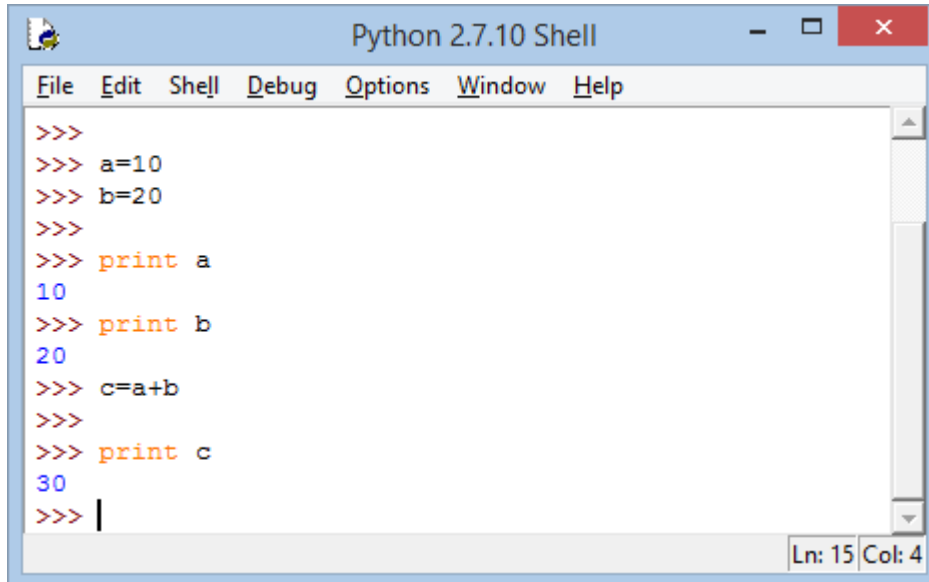


## Quick Overview to Python Core

**Note** – The guide is important who want to have a quick overview over the basics of python. Programs are written along with output and screenshot.

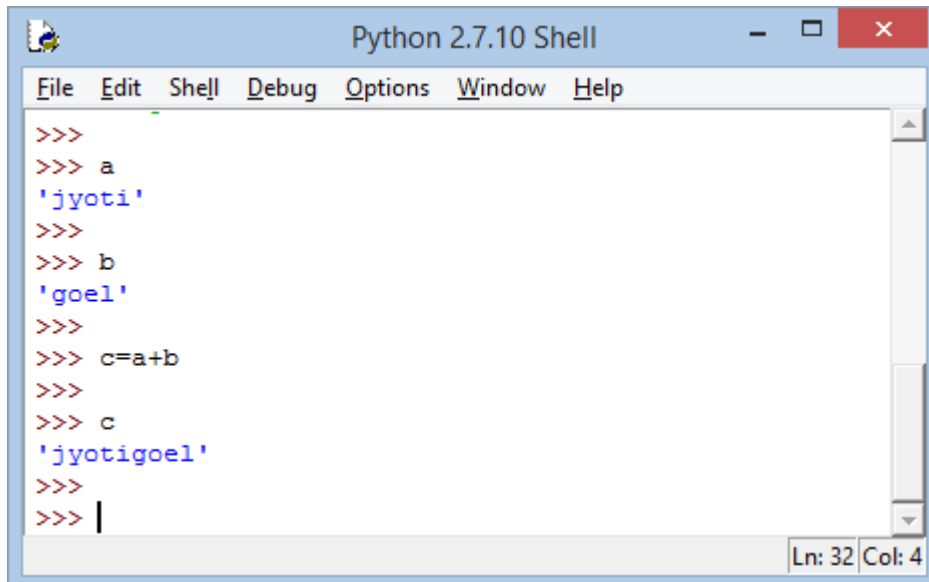
/\* Ques1. A program to add two numbers and print their sum. \*/

A screenshot of a Python 2.7.10 Shell window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area contains the following code:

```
>>>
>>> a=10
>>> b=20
>>>
>>> print a
10
>>> print b
20
>>> c=a+b
>>>
>>> print c
30
>>> |
```

The status bar at the bottom right shows 'Ln: 15 Col: 4'.

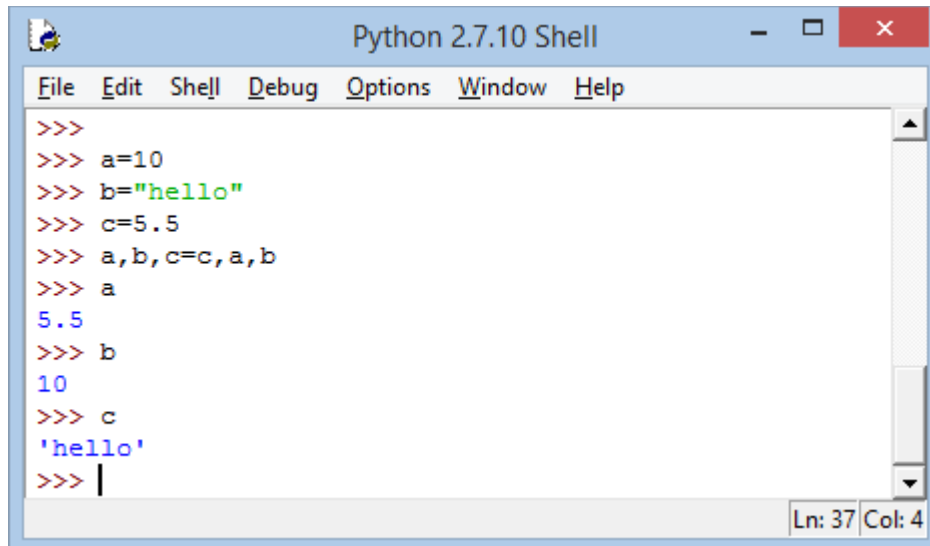
/\* Ques2. A program to take two strings and concatenate them \*/

A screenshot of a Python 2.7.10 Shell window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area contains the following code:

```
>>>
>>> a
'jyoti'
>>>
>>> b
'goel'
>>>
>>> c=a+b
>>>
>>> c
'jyotigoel'
>>>
>>> |
```

The status bar at the bottom right shows 'Ln: 32 Col: 4'.

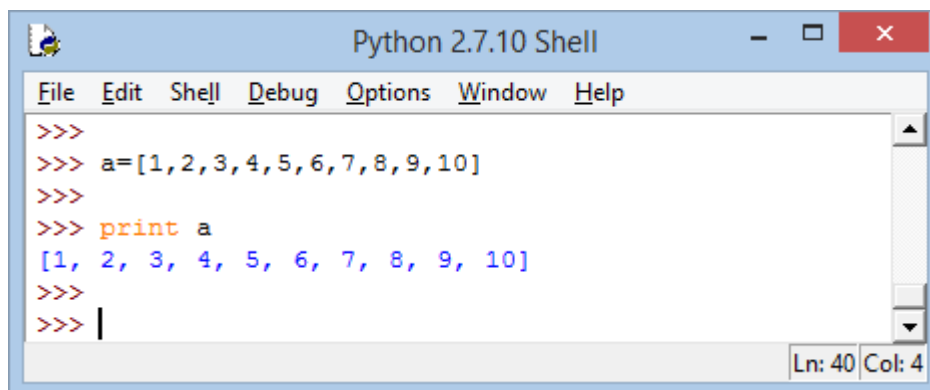
/\* Ques3. Program to swap an integer, a float and a character value. \*/



```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>>
>>> a=10
>>> b="hello"
>>> c=5.5
>>> a,b,c=c,a,b
>>> a
5.5
>>> b
10
>>> c
'hello'
>>> |
```

Ln: 37 Col: 4

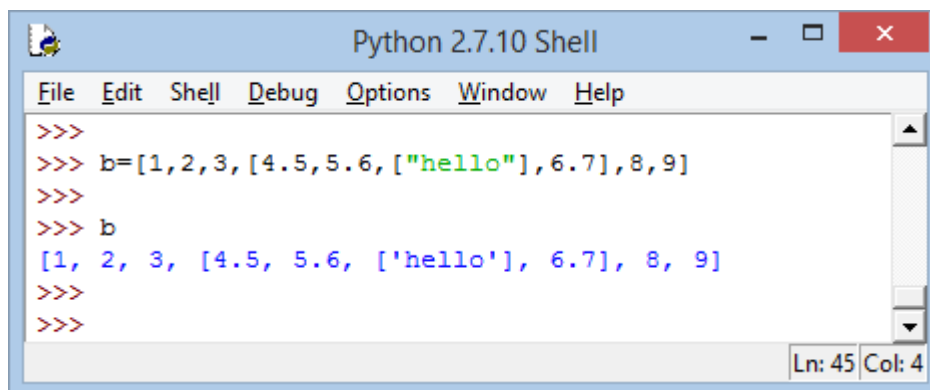
/\* Ques4. Create a python list from 1 to 10. \*/



```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>>
>>> a=[1,2,3,4,5,6,7,8,9,10]
>>>
>>> print a
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>>
>>> |
```

Ln: 40 Col: 4

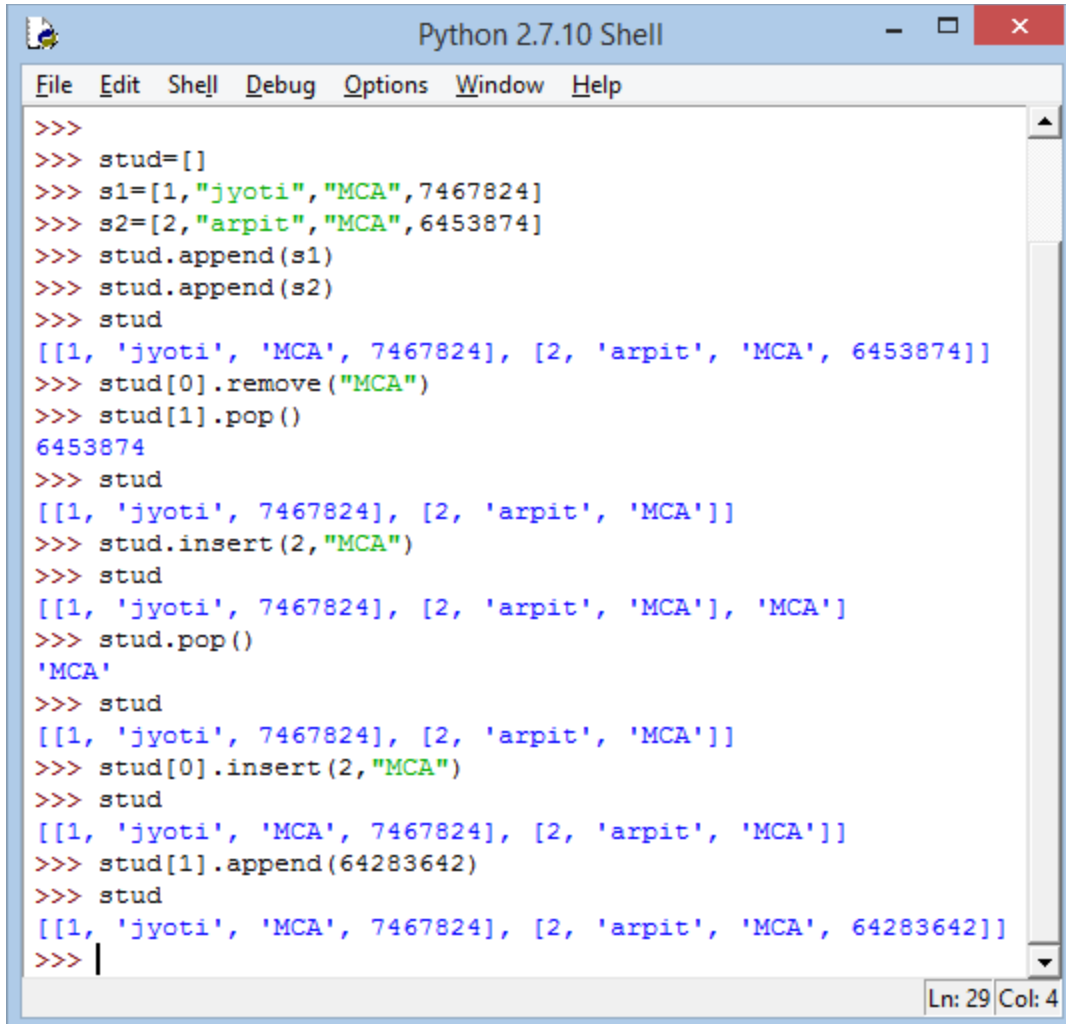
/\* Ques5. Create a nested list including numbers and characters. \*/



```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>>
>>> b=[1,2,3,[4.5,5.6,["hello"],6.7],8,9]
>>>
>>> b
[1, 2, 3, [4.5, 5.6, ['hello'], 6.7], 8, 9]
>>>
>>> |
```

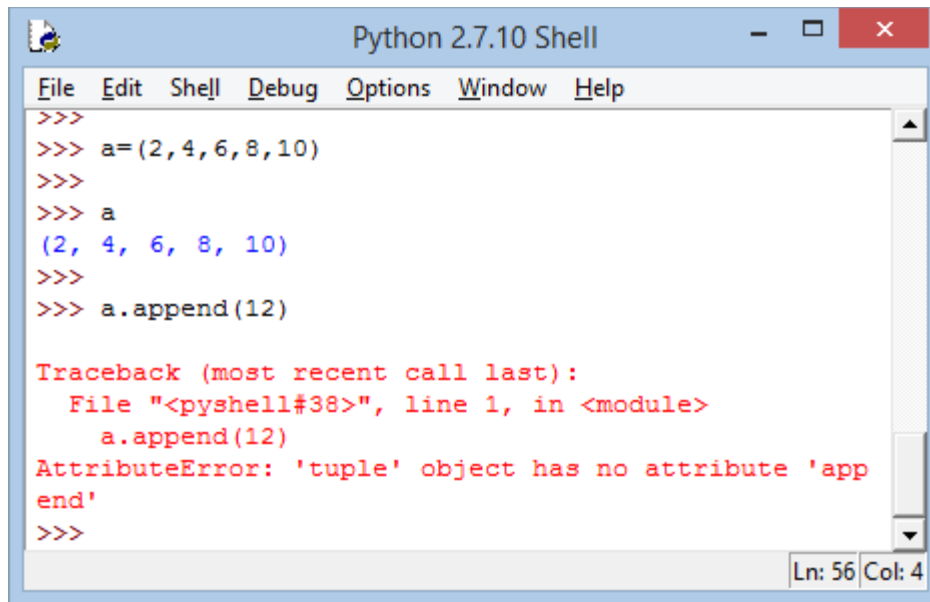
Ln: 45 Col: 4

/\* Ques6. Program to create and maintain students record. \*/

A screenshot of a Python 2.7.10 Shell window. The window has a title bar with the text "Python 2.7.10 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with options: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window contains a series of Python commands and their outputs. The commands are: 1. >>> stud=[] 2. >>> s1=[1,"jyoti","MCA",7467824] 3. >>> s2=[2,"arpit","MCA",6453874] 4. >>> stud.append(s1) 5. >>> stud.append(s2) 6. >>> stud 7. [[1, 'jyoti', 'MCA', 7467824], [2, 'arpit', 'MCA', 6453874]] 8. >>> stud[0].remove("MCA") 9. >>> stud[1].pop() 10. 6453874 11. >>> stud 12. [[1, 'jyoti', 7467824], [2, 'arpit', 'MCA']] 13. >>> stud.insert(2,"MCA") 14. >>> stud 15. [[1, 'jyoti', 7467824], [2, 'arpit', 'MCA'], 'MCA'] 16. >>> stud.pop() 17. 'MCA' 18. >>> stud 19. [[1, 'jyoti', 7467824], [2, 'arpit', 'MCA']] 20. >>> stud[0].insert(2,"MCA") 21. >>> stud 22. [[1, 'jyoti', 'MCA', 7467824], [2, 'arpit', 'MCA']] 23. >>> stud[1].append(64283642) 24. >>> stud 25. [[1, 'jyoti', 'MCA', 7467824], [2, 'arpit', 'MCA', 64283642]] 26. >>> | The status bar at the bottom right shows "Ln: 29 Col: 4".

```
>>>
>>> stud=[]
>>> s1=[1,"jyoti","MCA",7467824]
>>> s2=[2,"arpit","MCA",6453874]
>>> stud.append(s1)
>>> stud.append(s2)
>>> stud
[[1, 'jyoti', 'MCA', 7467824], [2, 'arpit', 'MCA', 6453874]]
>>> stud[0].remove("MCA")
>>> stud[1].pop()
6453874
>>> stud
[[1, 'jyoti', 7467824], [2, 'arpit', 'MCA']]
>>> stud.insert(2,"MCA")
>>> stud
[[1, 'jyoti', 7467824], [2, 'arpit', 'MCA'], 'MCA']
>>> stud.pop()
'MCA'
>>> stud
[[1, 'jyoti', 7467824], [2, 'arpit', 'MCA']]
>>> stud[0].insert(2,"MCA")
>>> stud
[[1, 'jyoti', 'MCA', 7467824], [2, 'arpit', 'MCA']]
>>> stud[1].append(64283642)
>>> stud
[[1, 'jyoti', 'MCA', 7467824], [2, 'arpit', 'MCA', 64283642]]
>>> |
```

/\* Ques7. Create a tuple of 5 numbers. \*/

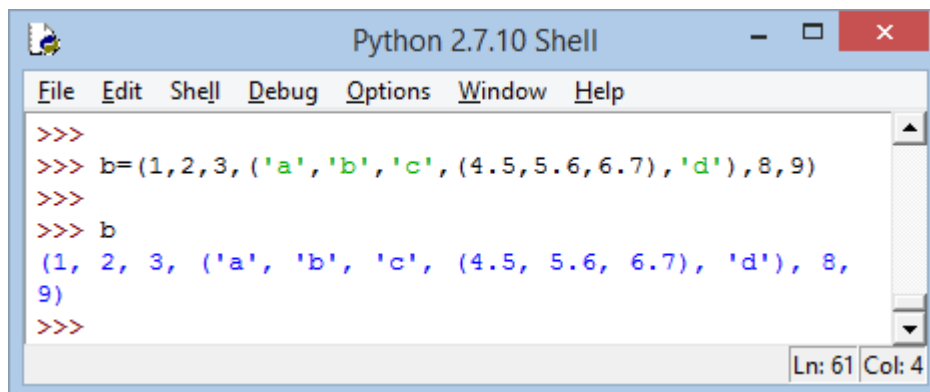


```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>>
>>> a=(2,4,6,8,10)
>>>
>>> a
(2, 4, 6, 8, 10)
>>>
>>> a.append(12)

Traceback (most recent call last):
  File "<pyshell#38>", line 1, in <module>
    a.append(12)
AttributeError: 'tuple' object has no attribute 'append'
>>>
```

Ln: 56 Col: 4

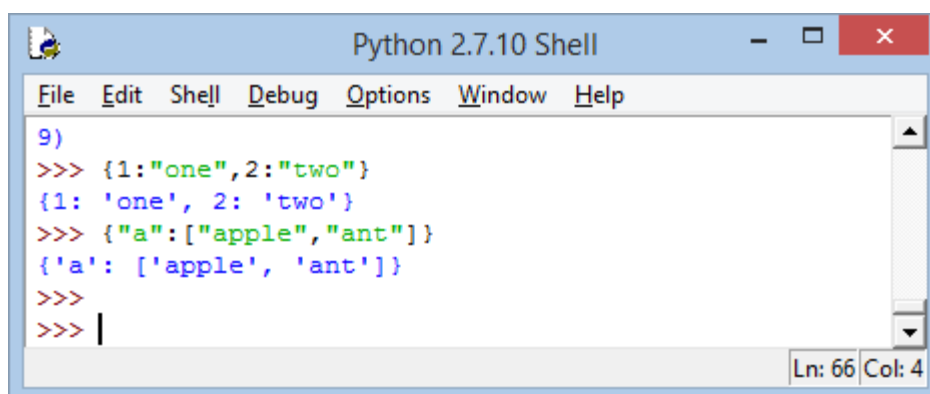
/\* Ques8. Create a nested tuple including numbers and characters. \*/



```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>>
>>> b=(1,2,3,('a','b','c',(4.5,5.6,6.7),'d'),8,9)
>>>
>>> b
(1, 2, 3, ('a', 'b', 'c', (4.5, 5.6, 6.7), 'd'), 8, 9)
>>>
```

Ln: 61 Col: 4

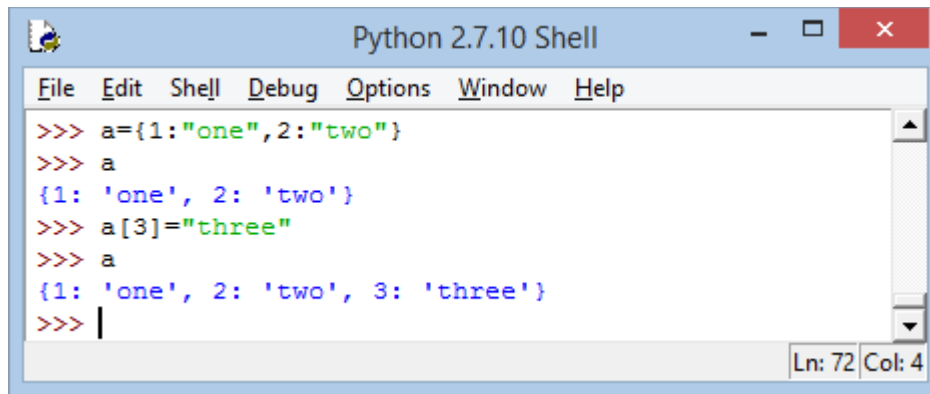
/\* Ques9. Create a dictionary. \*/



```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
9)
>>> {1:"one",2:"two"}
{1: 'one', 2: 'two'}
>>> {"a":["apple","ant"]}
{'a': ['apple', 'ant']}
>>>
>>> |
```

Ln: 66 Col: 4

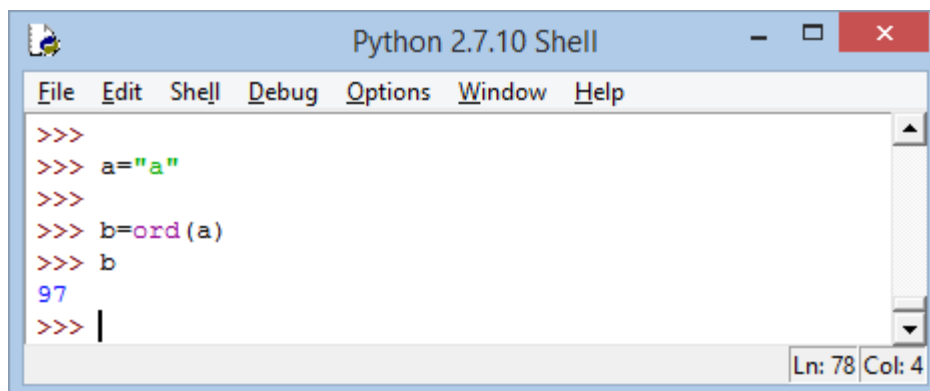
/\* Ques10. Add values in a pre-defined dictionary. \*/



```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>> a={1:"one",2:"two"}
>>> a
{1: 'one', 2: 'two'}
>>> a[3]="three"
>>> a
{1: 'one', 2: 'two', 3: 'three'}
>>> |
```

Ln: 72 Col: 4

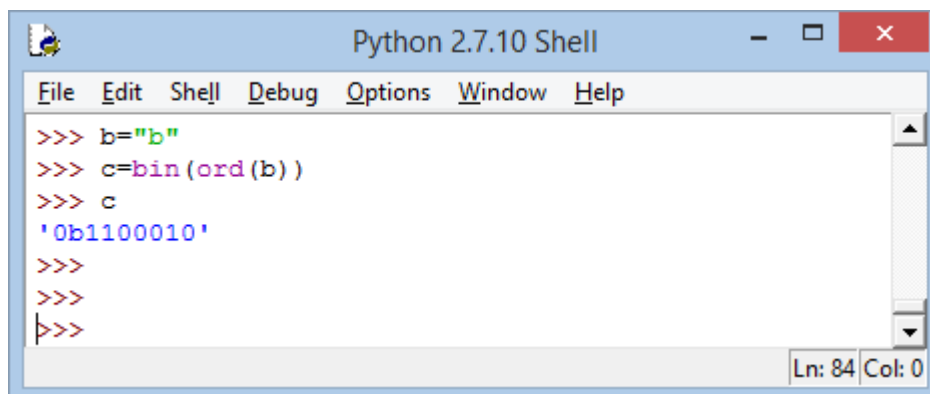
/\* Ques11. Print ASCII value of character.\*/



```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>>
>>> a="a"
>>>
>>> b=ord(a)
>>> b
97
>>> |
```

Ln: 78 Col: 4

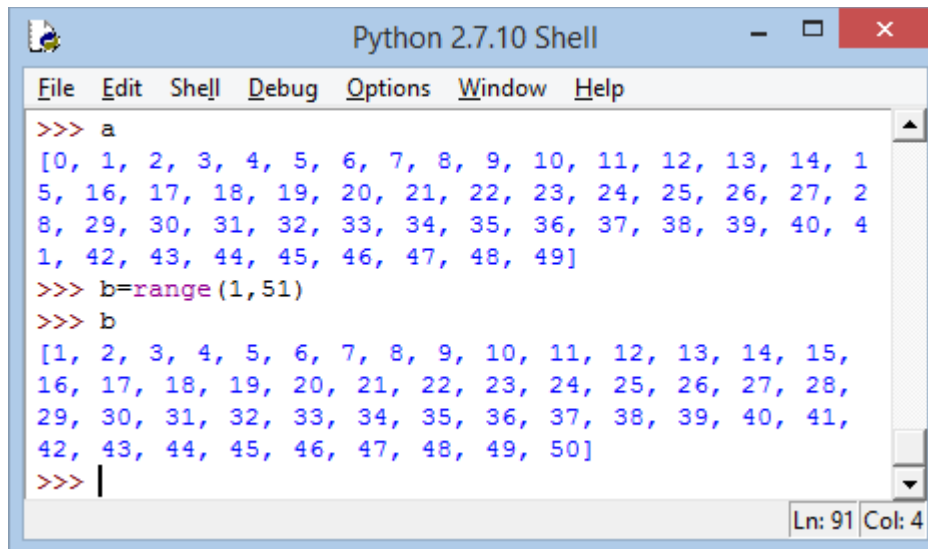
/\* Ques12. Print binary value of a character. \*/



```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>> b="b"
>>> c=bin(ord(b))
>>> c
'0b1100010'
>>>
>>>
>>> |
```

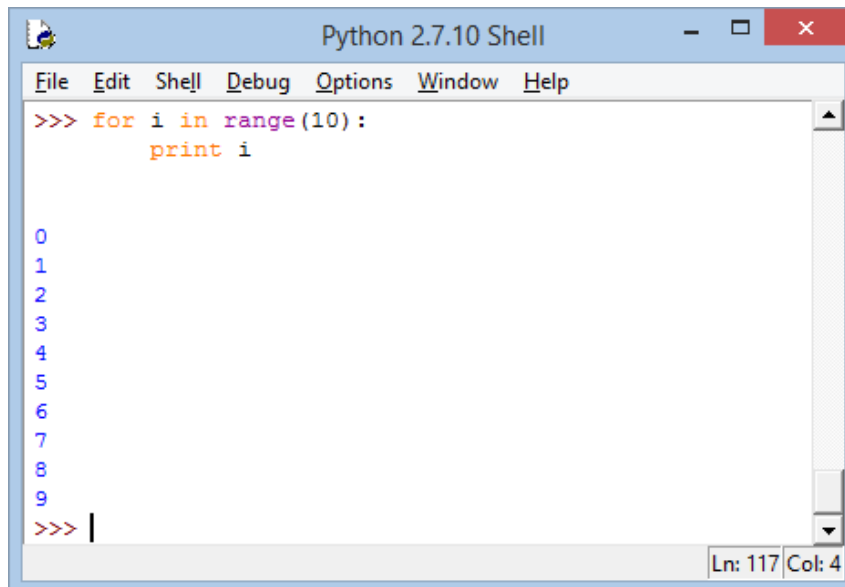
Ln: 84 Col: 0

/\* Ques13. Program to print 50 numbers using range() function. \*/



```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>> a
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1
5, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 2
8, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 4
1, 42, 43, 44, 45, 46, 47, 48, 49]
>>> b=range(1,51)
>>> b
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28,
29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
42, 43, 44, 45, 46, 47, 48, 49, 50]
>>> |
Ln: 91 Col: 4
```

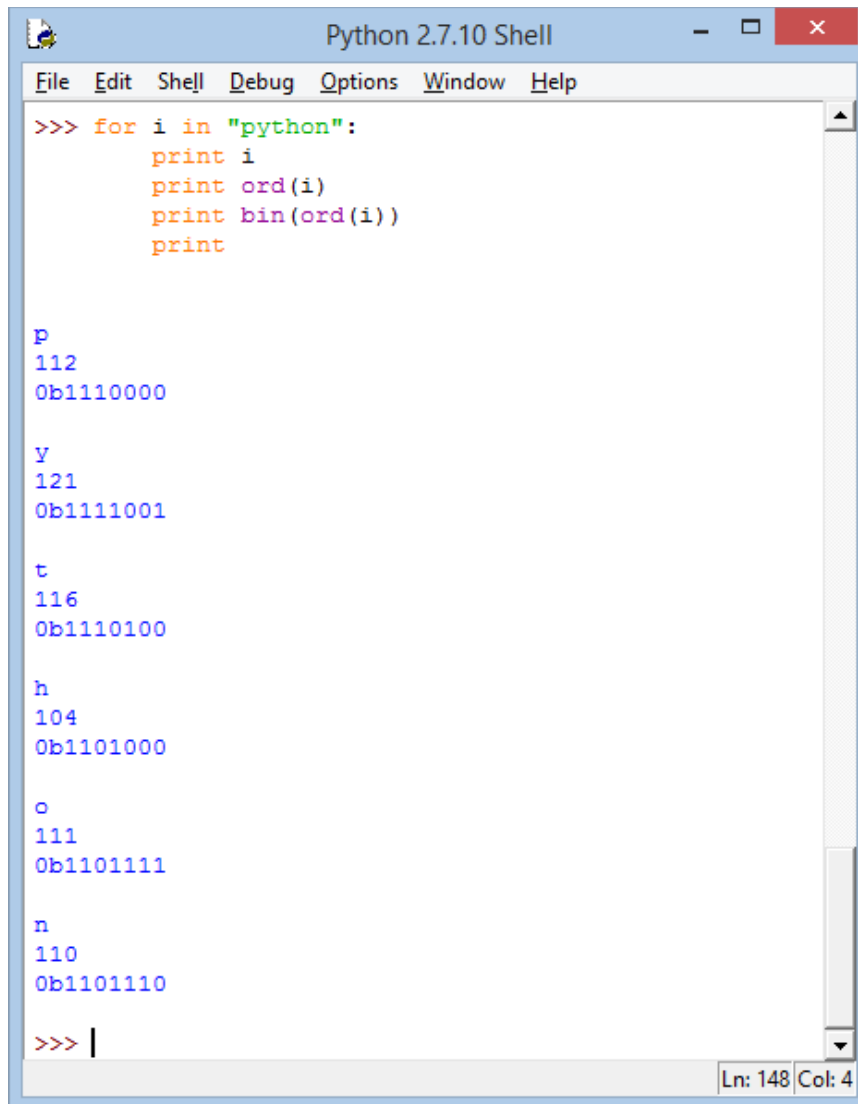
/\* Ques14. Program to print 10 numbers using for loop. \*/



```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>> for i in range(10):
    print i

0
1
2
3
4
5
6
7
8
9
>>> |
Ln: 117 Col: 4
```

/\*Ques15. Program to print ASCII and binary value of a string character by character. \*/



```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>> for i in "python":
    print i
    print ord(i)
    print bin(ord(i))
    print

p
112
0b1110000

y
121
0b1111001

t
116
0b1110100

h
104
0b1101000

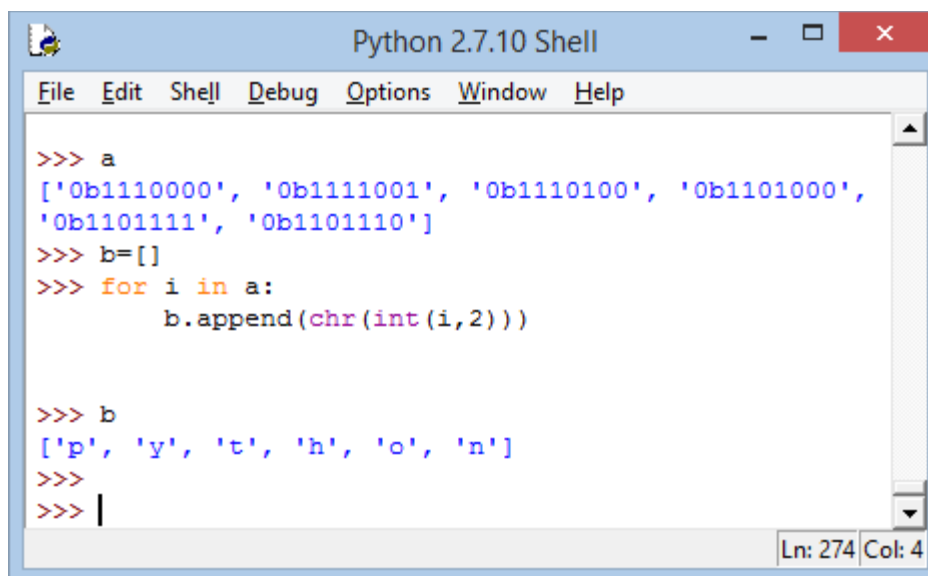
o
111
0b1101111

n
110
0b1101110

>>> |
```

Ln: 148 Col: 4

/\* Ques16. Program to convert binary list into a characters list. \*/

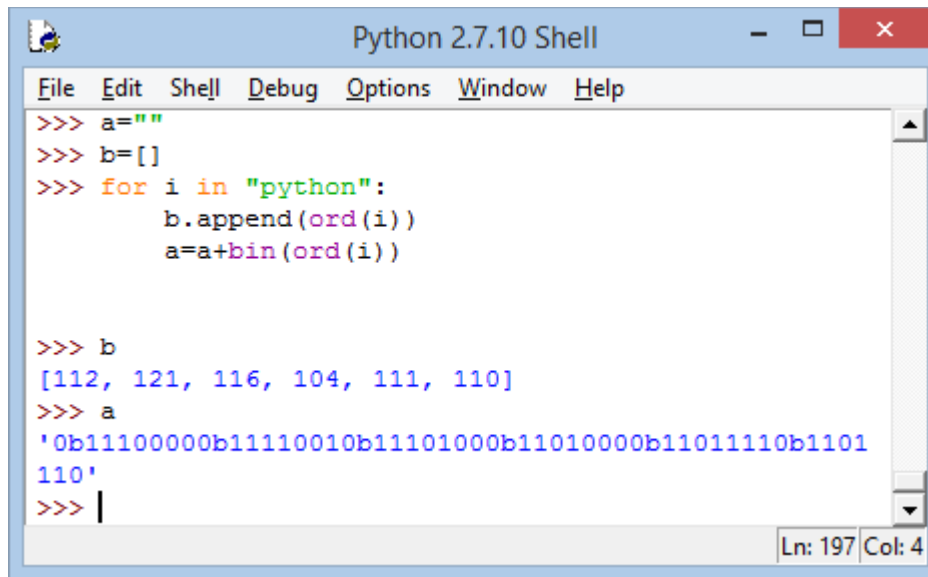


```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>> a
['0b1110000', '0b1111001', '0b1110100', '0b1101000',
'0b1101111', '0b1101110']
>>> b=[]
>>> for i in a:
    b.append(chr(int(i,2)))

>>> b
['p', 'y', 't', 'h', 'o', 'n']
>>>
>>> |
```

Ln: 274 Col: 4

/\* Ques17. Create an ASCII list and a binary string of a given string. \*/

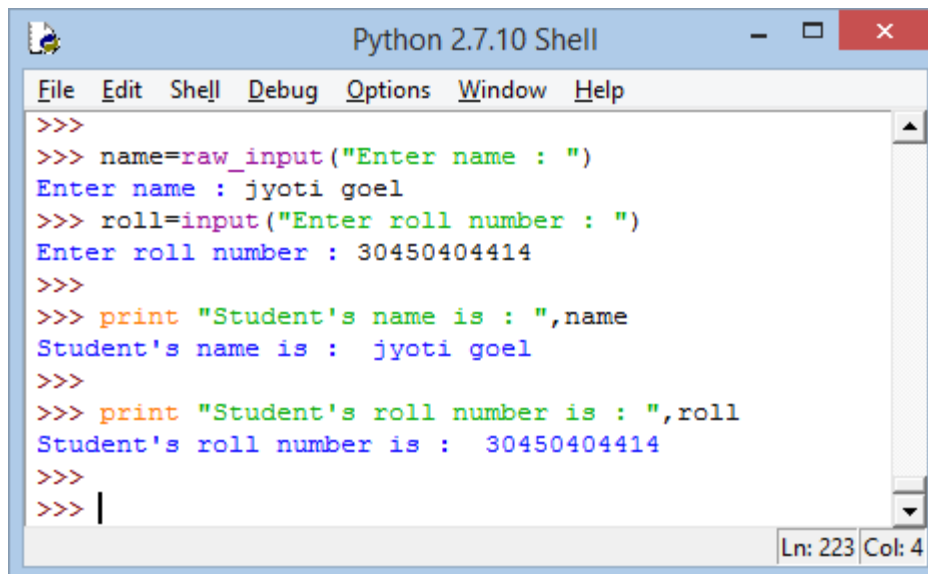


```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>> a=""
>>> b=[]
>>> for i in "python":
>>>     b.append(ord(i))
>>>     a=a+bin(ord(i))

>>> b
[112, 121, 116, 104, 111, 110]
>>> a
'0b11100000b11110010b11101000b11010000b11011110b1101110'
>>> |
```

Ln: 197 Col: 4

/\* Ques18. Take name and roll number as an input from a student. \*/

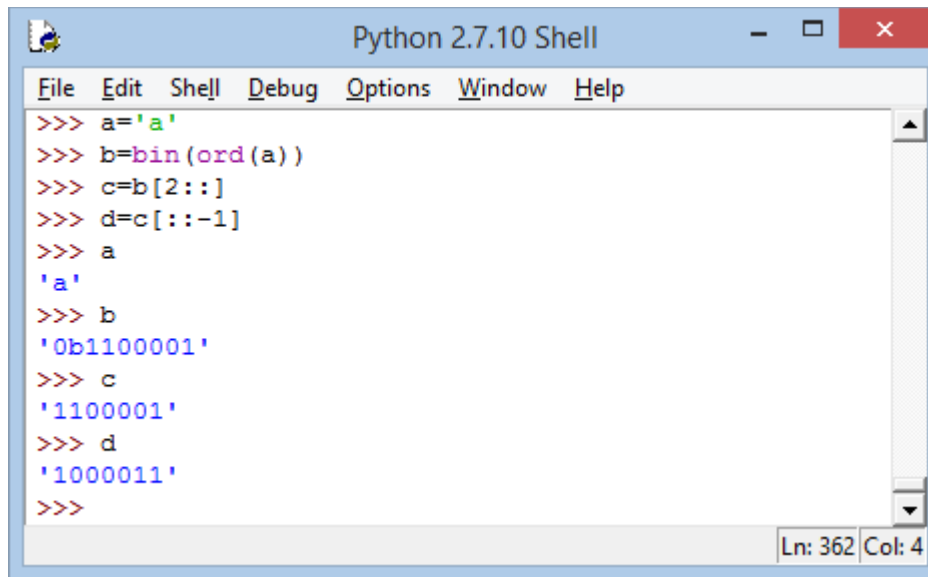


```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>>
>>> name=raw_input("Enter name : ")
Enter name : jyoti goel
>>> roll=input("Enter roll number : ")
Enter roll number : 30450404414
>>>
>>> print "Student's name is : ",name
Student's name is :  jyoti goel
>>>
>>> print "Student's roll number is : ",roll
Student's roll number is :  30450404414
>>>
>>> |
```

Ln: 223 Col: 4

/\* Ques19. Program to print binary value of a character and then print it in reverse order. \*/

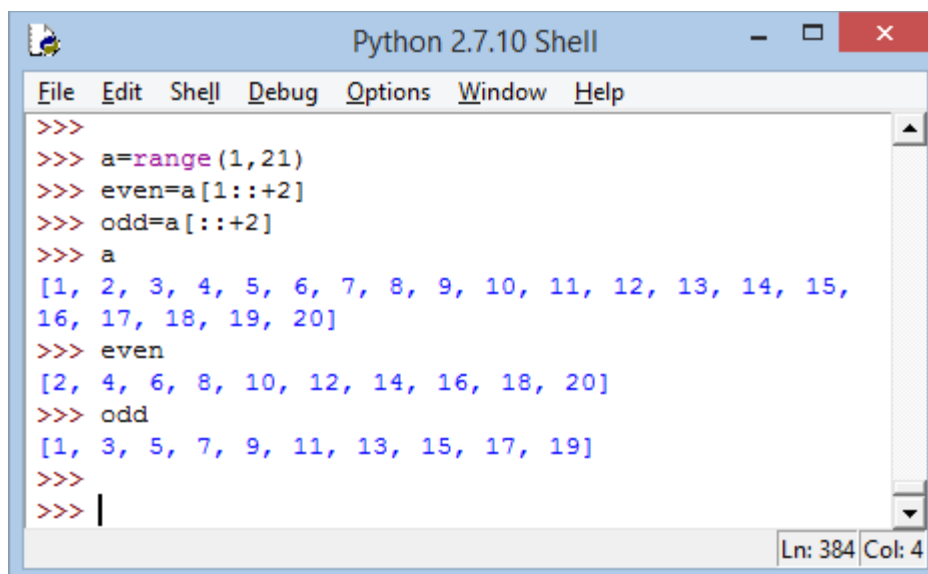




```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>> a='a'
>>> b=bin(ord(a))
>>> c=b[2::]
>>> d=c[::-1]
>>> a
'a'
>>> b
'0b1100001'
>>> c
'1100001'
>>> d
'1000011'
>>>
```

Ln: 362 Col: 4

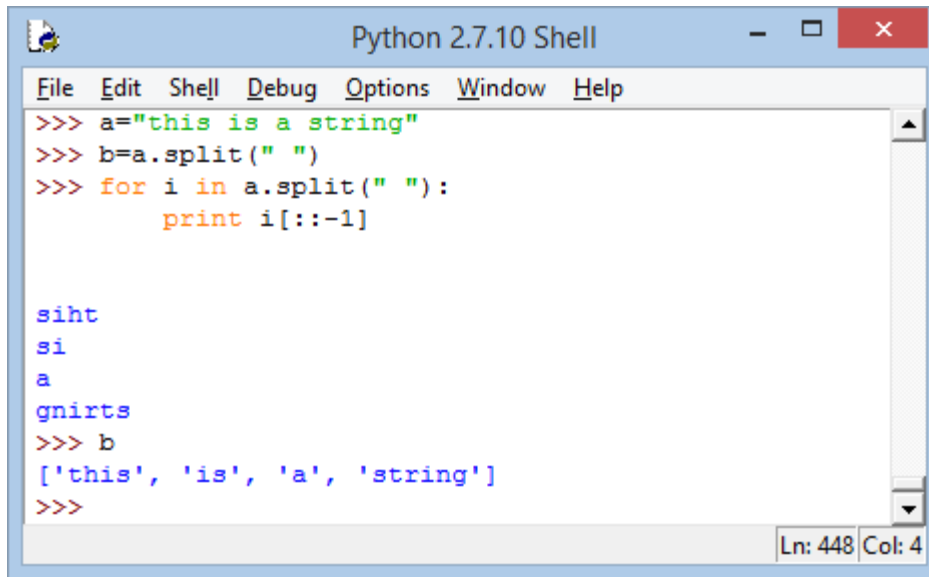
/\* Ques20. Print a list containing numbers from 1 to 20. Then print even and odd numbers from this list separately. \*/



```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>>
>>> a=range(1,21)
>>> even=a[1::+2]
>>> odd=a[::+2]
>>> a
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
>>> even
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
>>> odd
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
>>>
>>> |
```

Ln: 384 Col: 4

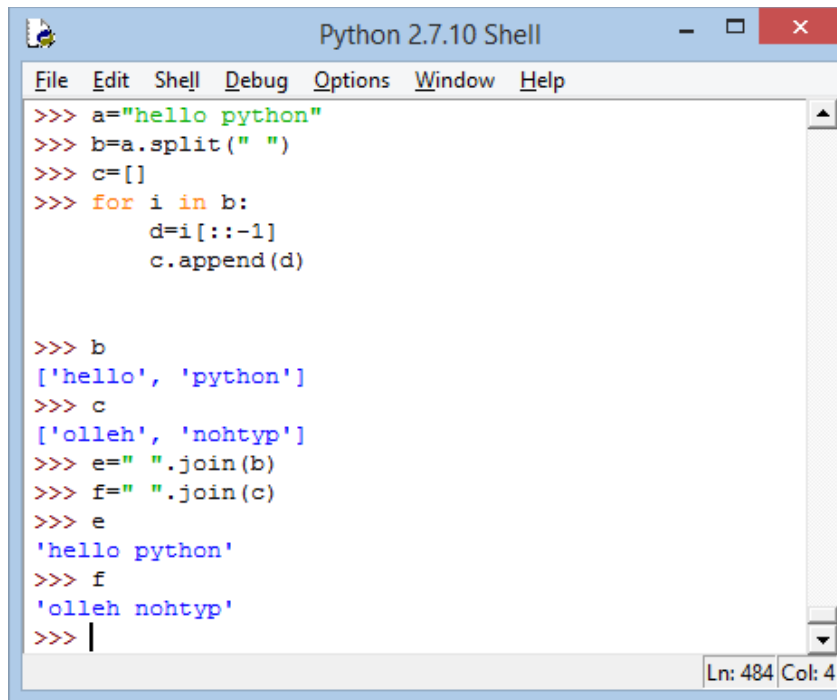
/\* Ques21. Program to split a string into words and print the individual word in reverse order.\*/

A screenshot of a Python 2.7.10 Shell window. The window has a title bar with the text "Python 2.7.10 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with options: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window contains a Python script. The script starts with a prompt ">>>" followed by the assignment "a='this is a string'". Then another prompt ">>>" followed by "b=a.split(' ')", and a third prompt ">>>" followed by a for loop: "for i in a.split(' '):" with an indented "print i[::-1]" statement. The output of the script is shown below the code: "siht", "si", "a", and "gnirts", each on a new line. Below the output, the prompt ">>>" is followed by the variable "b" and its value as a list: "['this', 'is', 'a', 'string']". The prompt ">>>" appears again on the next line. At the bottom right of the window, there is a status bar showing "Ln: 448" and "Col: 4".

```
>>> a="this is a string"
>>> b=a.split(" ")
>>> for i in a.split(" "):
    print i[::-1]

siht
si
a
gnirts
>>> b
['this', 'is', 'a', 'string']
>>>
```

/\* Ques22. Program to split a string, reverse each word and join them to form a reversed string. \*/

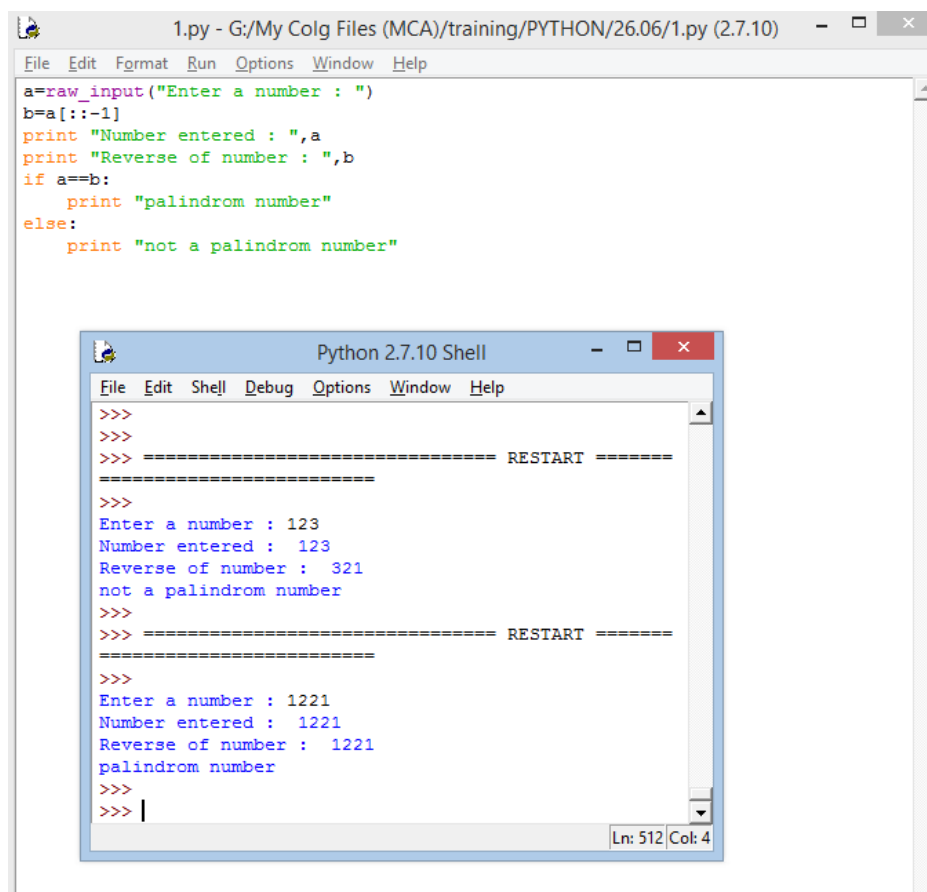


```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>> a="hello python"
>>> b=a.split(" ")
>>> c=[]
>>> for i in b:
>>>     d=i[::-1]
>>>     c.append(d)

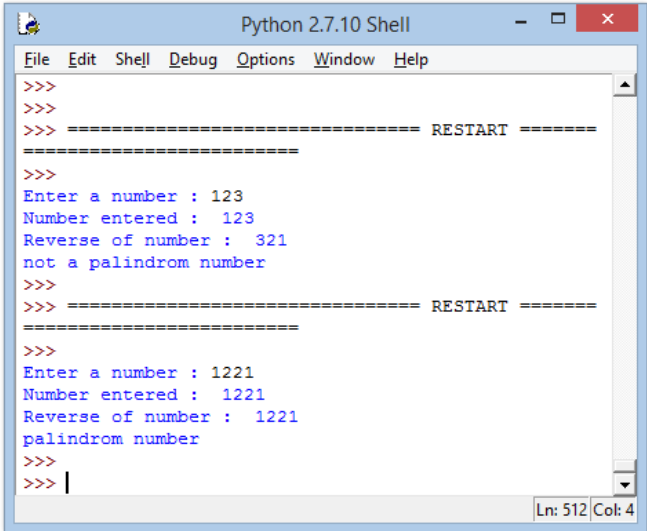
>>> b
['hello', 'python']
>>> c
['olleh', 'nohtyp']
>>> e=" ".join(b)
>>> f=" ".join(c)
>>> e
'hello python'
>>> f
'olleh nohtyp'
>>> |
```

Ln: 484 Col: 4

/\* Ques23. Program to find palindrome number using if else statements.\*/



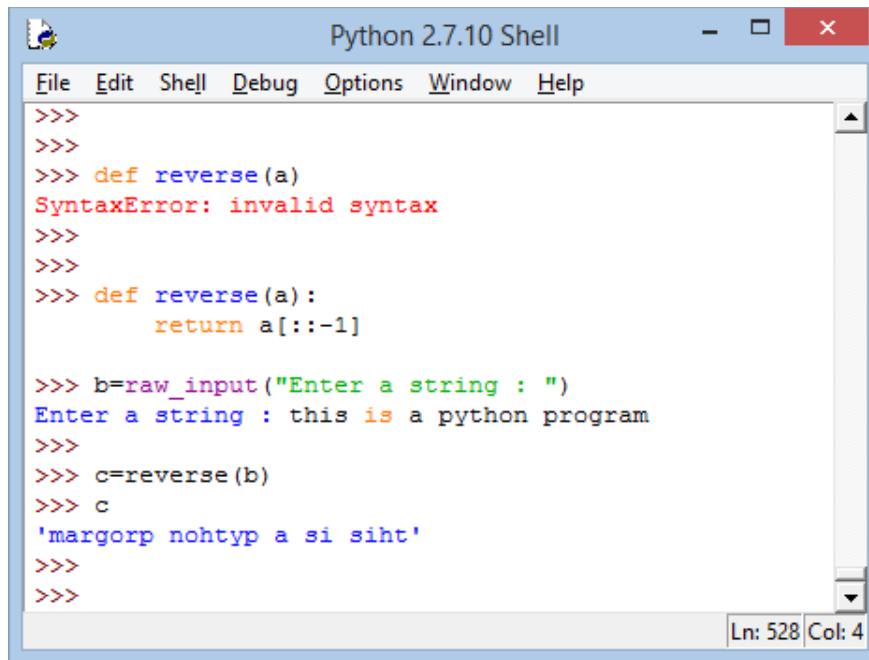
```
1.py - G:/My Colg Files (MCA)/training/PYTHON/26.06/1.py (2.7.10)
File Edit Format Run Options Window Help
a=raw_input("Enter a number : ")
b=a[::-1]
print "Number entered : ",a
print "Reverse of number : ",b
if a==b:
    print "palindrom number"
else:
    print "not a palindrom number"
```



```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>>
>>>
>>> ===== RESTART =====
>>>
>>> Enter a number : 123
>>> Number entered : 123
>>> Reverse of number : 321
>>> not a palindrom number
>>>
>>> ===== RESTART =====
>>>
>>> Enter a number : 1221
>>> Number entered : 1221
>>> Reverse of number : 1221
>>> palindrom number
>>>
>>> |
```

Ln: 512 Col: 4

/\* Ques24. Program to reverse a string using function. \*/

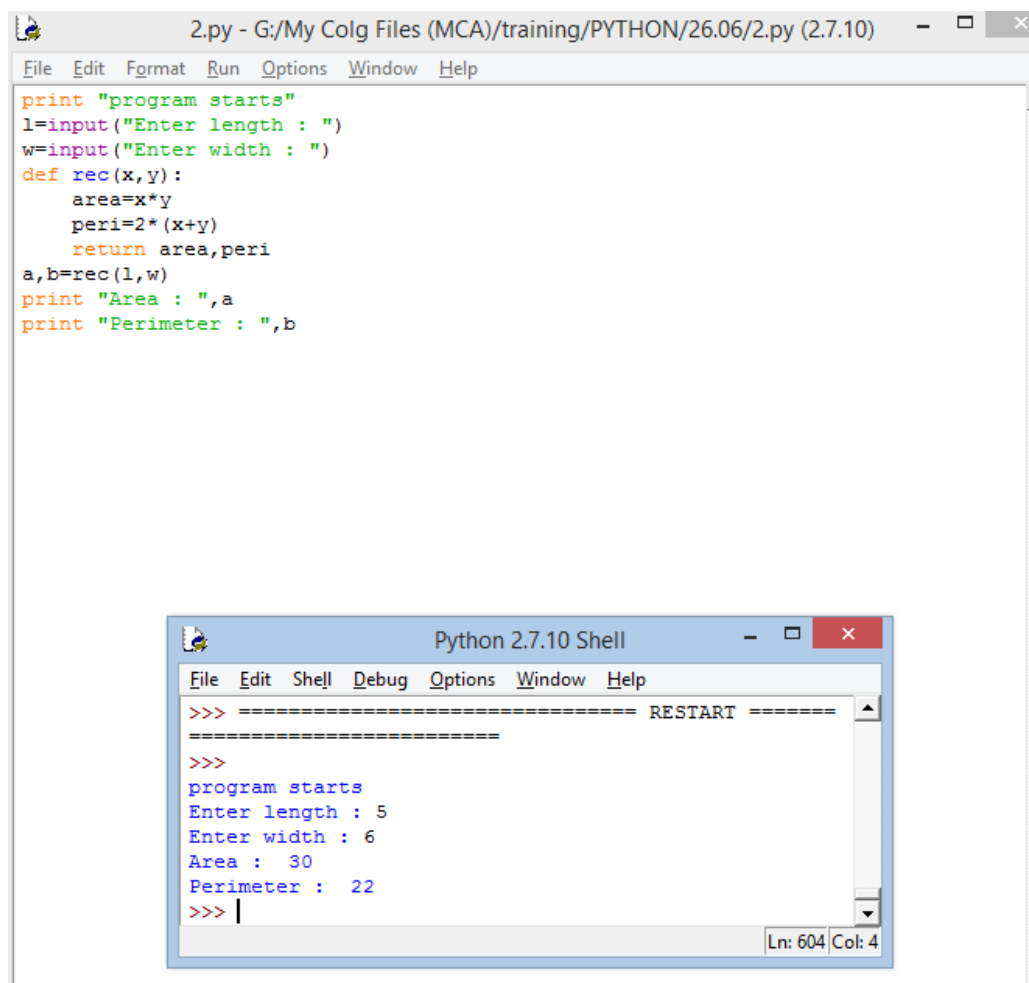


```
>>>
>>>
>>> def reverse(a)
SyntaxError: invalid syntax
>>>
>>>
>>> def reverse(a):
        return a[::-1]

>>> b=raw_input("Enter a string : ")
Enter a string : this is a python program
>>>
>>> c=reverse(b)
>>> c
'margorp nohtyp a si siht'
>>>
>>>
```

Ln: 528 Col: 4

/\* Ques25. Program to create a function which produces area and perimeter of a rectangle and return both. \*/

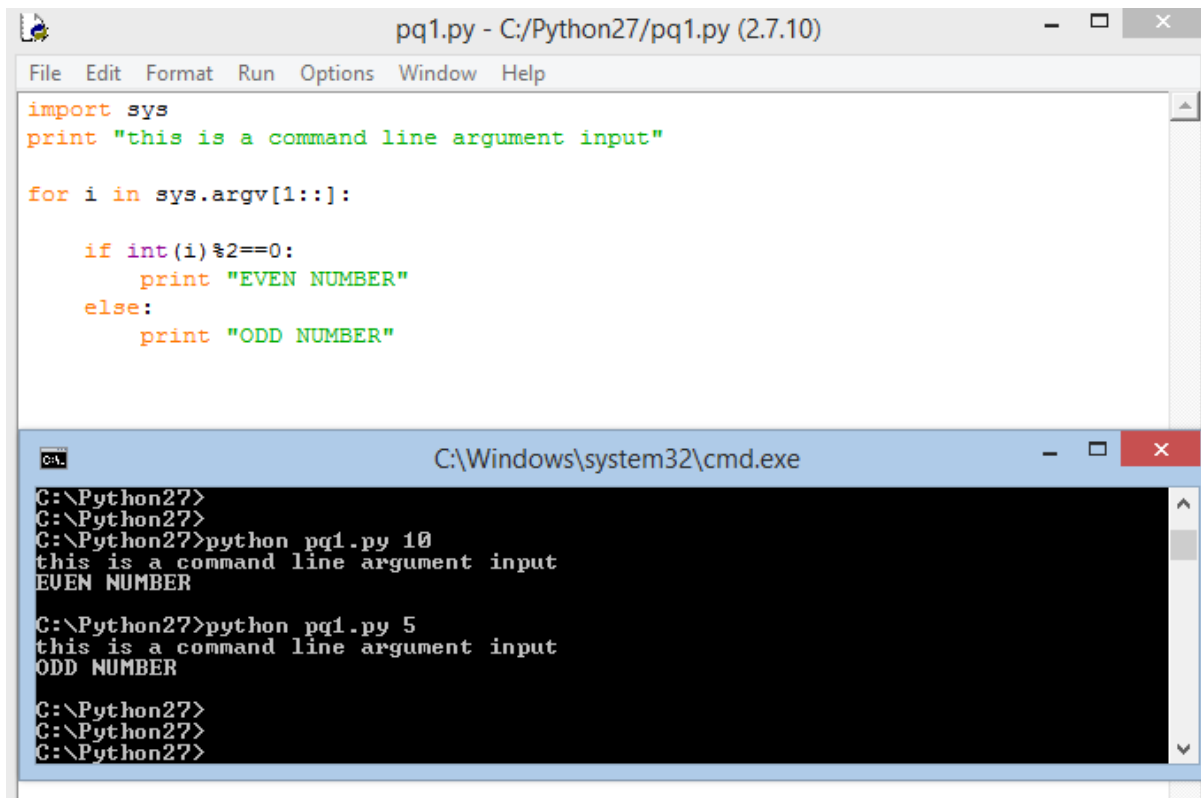


```
2.py - G:/My Colg Files (MCA)/training/PYTHON/26.06/2.py (2.7.10)
File Edit Format Run Options Window Help
print "program starts"
l=input("Enter length : ")
w=input("Enter width : ")
def rec(x,y):
    area=x*y
    peri=2*(x+y)
    return area,peri
a,b=rec(l,w)
print "Area : ",a
print "Perimeter : ",b

Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>> ===== RESTART =====
>>>
>>> program starts
>>> Enter length : 5
>>> Enter width : 6
>>> Area : 30
>>> Perimeter : 22
>>> |
```

Ln: 604 Col: 4

/\* Ques26. Program to find number entered is even or odd. Input is a command line argument. \*/



The screenshot shows two windows. The top window is a Python IDE titled 'pq1.py - C:/Python27/pq1.py (2.7.10)'. It contains the following code:

```
import sys
print "this is a command line argument input"

for i in sys.argv[1::]:
    if int(i)%2==0:
        print "EVEN NUMBER"
    else:
        print "ODD NUMBER"
```

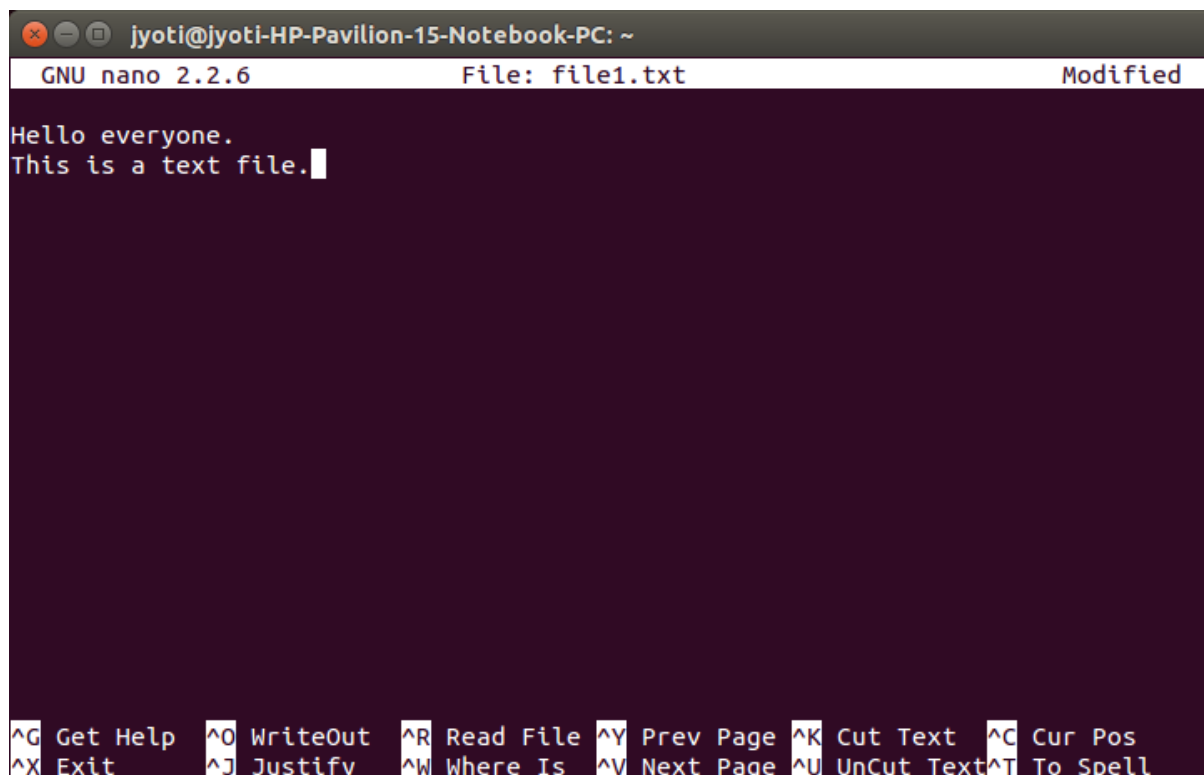
The bottom window is a command prompt titled 'C:\Windows\system32\cmd.exe'. It shows the execution of the program with two different inputs:

```
C:\Python27>
C:\Python27>
C:\Python27>python pq1.py 10
this is a command line argument input
EVEN NUMBER

C:\Python27>python pq1.py 5
this is a command line argument input
ODD NUMBER

C:\Python27>
C:\Python27>
C:\Python27>
```

/\* Ques27. Program to read from a file using read() function. \*/



The screenshot shows a terminal window with the nano text editor. The title bar indicates the user is 'jyoti@jyoti-HP-Pavilion-15-Notebook-PC: ~' and the file being edited is 'file1.txt'. The editor shows the following text:

```
Hello everyone.
This is a text file.
```

The bottom status bar of the nano editor displays various keyboard shortcuts for navigation and editing, such as '^G Get Help', '^O WriteOut', '^R Read File', etc.

```

jyoti@jyoti-HP-Pavilion-15-Notebook-PC: ~
jyoti@jyoti-HP-Pavilion-15-Notebook-PC:~$ pwd
/home/jyoti
jyoti@jyoti-HP-Pavilion-15-Notebook-PC:~$ ls
Desktop  Downloads      Music  Public  Templates
Documents examples.desktop Pictures python Videos
jyoti@jyoti-HP-Pavilion-15-Notebook-PC:~$ touch file1.txt
jyoti@jyoti-HP-Pavilion-15-Notebook-PC:~$ ls
Desktop  Downloads      file1.txt  Pictures  python  Videos
Documents examples.desktop Music      Public    Templates
jyoti@jyoti-HP-Pavilion-15-Notebook-PC:~$ nano file1.txt
jyoti@jyoti-HP-Pavilion-15-Notebook-PC:~$ python
Python 2.7.8 (default, Oct 20 2014, 15:05:19)
[GCC 4.9.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> f=open("file1.txt","r")
>>> f
<open file 'file1.txt', mode 'r' at 0x7fbcc5f9e5d0>
>>> f.read()
'Hello everyone.\nThis is a text file.\n'
>>> f.close()
>>> 

```

/\* Ques28. Program to open a file using with. \*/

```

jyoti@jyoti-HP-Pavilion-15-Notebook-PC: ~
jyoti@jyoti-HP-Pavilion-15-Notebook-PC:~$ ls
Desktop  Downloads      file1.txt  Pictures  python  Videos
Documents examples.desktop Music      Public    Templates
jyoti@jyoti-HP-Pavilion-15-Notebook-PC:~$ python
Python 2.7.8 (default, Oct 20 2014, 15:05:19)
[GCC 4.9.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 
>>> with open("file1.txt","r") as f:
...     f.read()
...
'Hello everyone.\nThis is a text file.\n'
>>> 

```

/\* Ques29. Program to read a file using readline() function. \*/

```
jyoti@jyoti-HP-Pavilion-15-Notebook-PC: ~
jyoti@jyoti-HP-Pavilion-15-Notebook-PC:~$ python
Python 2.7.8 (default, Oct 20 2014, 15:05:19)
[GCC 4.9.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> f=open("file1.txt","r")
>>> f
<open file 'file1.txt', mode 'r' at 0x7f9e7c47d5d0>
>>> f.readline()
'Hello everyone.\n'
>>> f.readline()
'This is a text file.\n'
>>> f.close()
>>> 
```

/\*Ques30. Program to read a file using readlines() function. \*/

```
jyoti@jyoti-HP-Pavilion-15-Notebook-PC: ~
jyoti@jyoti-HP-Pavilion-15-Notebook-PC:~$ python
Python 2.7.8 (default, Oct 20 2014, 15:05:19)
[GCC 4.9.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> f=open("file1.txt","r")
>>> f
<open file 'file1.txt', mode 'r' at 0x7f94a7d565d0>
>>> f.readlines()
['Hello everyone.\n', 'This is a text file.\n']
>>> f.close()
>>> 
```

/\* Ques31. Program to write in a file. \*/

```

jyoti@jyoti-HP-Pavilion-15-Notebook-PC: ~
jyoti@jyoti-HP-Pavilion-15-Notebook-PC:~$ touch file2.txt
jyoti@jyoti-HP-Pavilion-15-Notebook-PC:~$ ls
Desktop    Downloads    file1.txt    Music        Public    Templates
Documents  examples.desktop  file2.txt    Pictures    python    Videos
jyoti@jyoti-HP-Pavilion-15-Notebook-PC:~$ python
Python 2.7.8 (default, Oct 20 2014, 15:05:19)
[GCC 4.9.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> f=open("file2.txt","w")
>>> f.write("This file is written using write mode.")
>>> f.close()
>>> f=open("file2.txt","r")
>>> f
<open file 'file2.txt', mode 'r' at 0x7f1d7c798660>
>>> f.read()
'This file is written using write mode.'
>>> f.close()
>>> 
```

/\* Ques32. Program to demonstrate exception handling. \*/

```

jyoti@jyoti-HP-Pavilion-15-Notebook-PC: ~
jyoti@jyoti-HP-Pavilion-15-Notebook-PC:~$ python
Python 2.7.8 (default, Oct 20 2014, 15:05:19)
[GCC 4.9.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> a=10
>>>
>>> try:
...     a/0
... except:
...     print "Exception : division by zero."
...
Exception : division by zero.
>>> 
```



/\* Ques33. Program to demonstrate the use of finally. \*/

```
jyoti@jyoti-HP-Pavilion-15-Notebook-PC: ~
>>> a=10
>>> b=input("Enter number : ")
Enter number : 0
>>> try:
...     print a/b
... except:
...     print "Exception : Division by zero."
... finally:
...     print "END"
...
Exception : Division by zero.
END
>>> c=input()
2
>>> try:
...     print a/c
... except:
...     print "Exception"
... finally:
...     print "END"
...
5
END
>>>
```

/\* Ques34. Program to create a dictionary which include the values as square of the key. \*/

```
jyoti@jyoti-HP-Pavilion-15-Notebook-PC: ~
>>>
>>> d={}
>>>
>>> for i in range(1,6):
...     d[i]=i*i,i**3
...
>>> d
{1: (1, 1), 2: (4, 8), 3: (9, 27), 4: (16, 64), 5: (25, 125)}
>>>
```

/\* Ques35. Program to print a list of all alphabetical characters using list comprehension. \*/

```
jyoti@jyoti-HP-Pavilion-15-Notebook-PC: ~
>>> [chr(x) for x in range(97,123)]
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p',
'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
>>>
>>>
>>> [chr(x) for x in range(65,91)]
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P',
'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
>>>
```

/\* Ques36. Program to print a binary string of a character string. Code should by of one line.  
\*/

```
jyoti@jyoti-HP-Pavilion-15-Notebook-PC: ~  
>>>  
>>> "".join("".join([bin(ord(x)) for x in "hello"]).split("0b"))  
'110100011001011110110011011001101111'  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>
```

/\* Ques37. Print a list of numbers based on certain condition of divisibility. \*/

```
jyoti@jyoti-HP-Pavilion-15-Notebook-PC: ~  
>>>  
>>> [x for x in range(1,1001) if x%6==0 and x%3!=0]  
[]  
>>>  
>>>  
>>> [x for x in range(1,1001) if x%6==0 and x%5!=0]  
[6, 12, 18, 24, 36, 42, 48, 54, 66, 72, 78, 84, 96, 102, 108, 114, 126, 132, 138  
, 144, 156, 162, 168, 174, 186, 192, 198, 204, 216, 222, 228, 234, 246, 252, 258  
, 264, 276, 282, 288, 294, 306, 312, 318, 324, 336, 342, 348, 354, 366, 372, 378  
, 384, 396, 402, 408, 414, 426, 432, 438, 444, 456, 462, 468, 474, 486, 492, 498  
, 504, 516, 522, 528, 534, 546, 552, 558, 564, 576, 582, 588, 594, 606, 612, 618  
, 624, 636, 642, 648, 654, 666, 672, 678, 684, 696, 702, 708, 714, 726, 732, 738  
, 744, 756, 762, 768, 774, 786, 792, 798, 804, 816, 822, 828, 834, 846, 852, 858  
, 864, 876, 882, 888, 894, 906, 912, 918, 924, 936, 942, 948, 954, 966, 972, 978  
, 984, 996]  
>>>
```

/\* Ques38. Program to demonstrate how to import a file. \*/

```
jyoti@jyoti-HP-Pavilion-15-Notebook-PC: ~  
>>>  
>>> import even_odd  
Enter number : 10  
EVEN NUMBER  
>>>  
>>> reload(even_odd)  
Enter number : 5  
ODD NUMBER  
<module 'even_odd' from 'even_odd.pyc'>  
>>>  
>>>
```

```
even_odd.py (~) - gedit
Open Save Undo
even_odd.py x
a=input("Enter number : ")
if a%2==0:
    print "EVEN NUMBER"
else:
    print "ODD NUMBER"
|

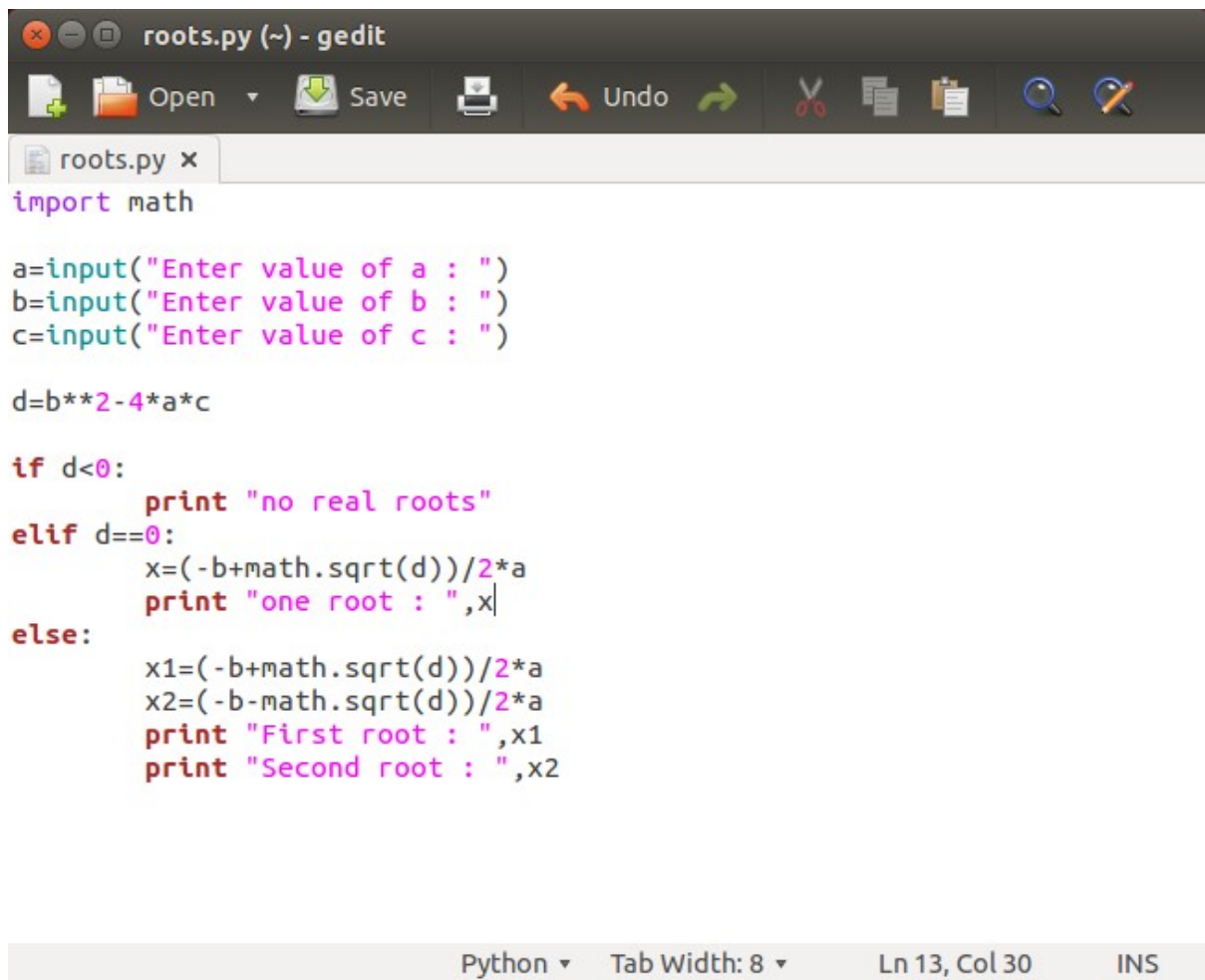
Python Tab Width: 8 Ln 6, Col 1 INS
```

/\* Ques39. Find square root of a number. \*/

```
jyoti@jyoti-HP-Pavilion-15-Notebook-PC: ~
>>>
>>> import math
>>> a=input("Enter number : ")
Enter number : 4
>>> math.sqrt(4)
2.0
>>>
>>> |
```

/\* Ques40. Program to solve a quadratic equation. \*/

```
jyoti@jyoti-HP-Pavilion-15-Notebook-PC: ~
>>> import roots
Enter value of a : 2
Enter value of b : 4
Enter value of c : 1
First root : -1.17157287525
Second root : -6.82842712475
>>>
>>> |
```



```
import math

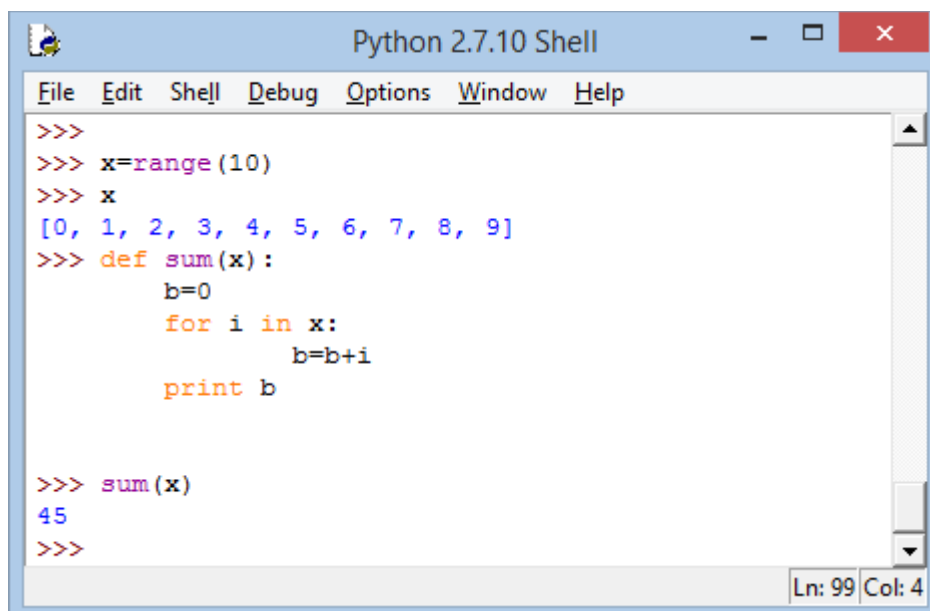
a=input("Enter value of a : ")
b=input("Enter value of b : ")
c=input("Enter value of c : ")

d=b**2-4*a*c

if d<0:
    print "no real roots"
elif d==0:
    x=(-b+math.sqrt(d))/2*a
    print "one root : ",x
else:
    x1=(-b+math.sqrt(d))/2*a
    x2=(-b-math.sqrt(d))/2*a
    print "First root : ",x1
    print "Second root : ",x2
```

Python ▾ Tab Width: 8 ▾ Ln 13, Col 30 INS

/\* Ques41. Function which take a list and print the sum of all the elements of list. \*/

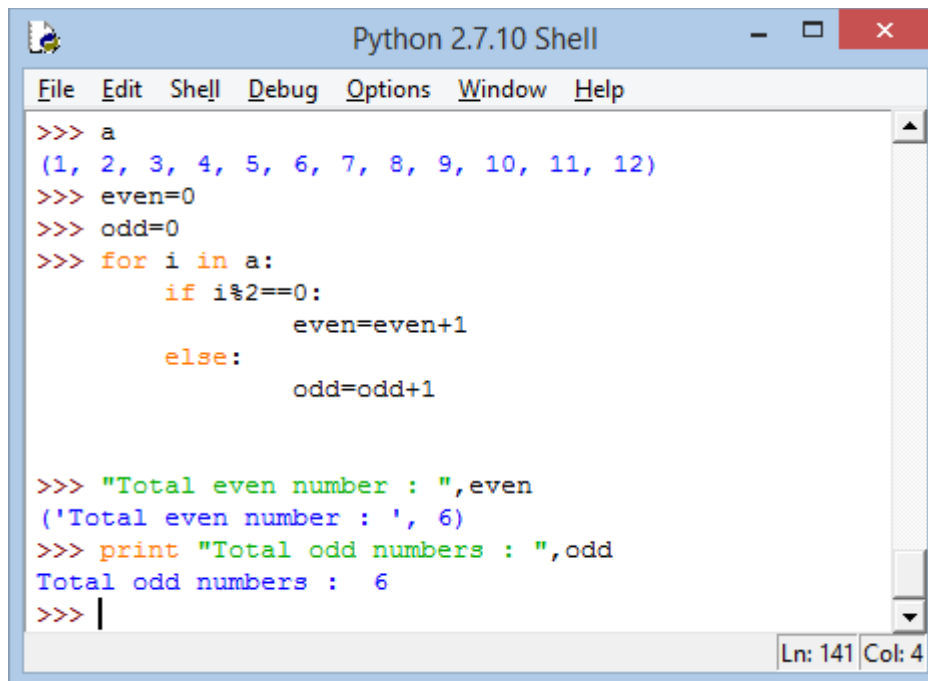


```
>>>
>>> x=range(10)
>>> x
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> def sum(x):
        b=0
        for i in x:
            b=b+i
        print b

>>> sum(x)
45
>>>
```

Ln: 99 Col: 4

/\* Ques42. Program to find total even and odd numbers in a tuple. \*/

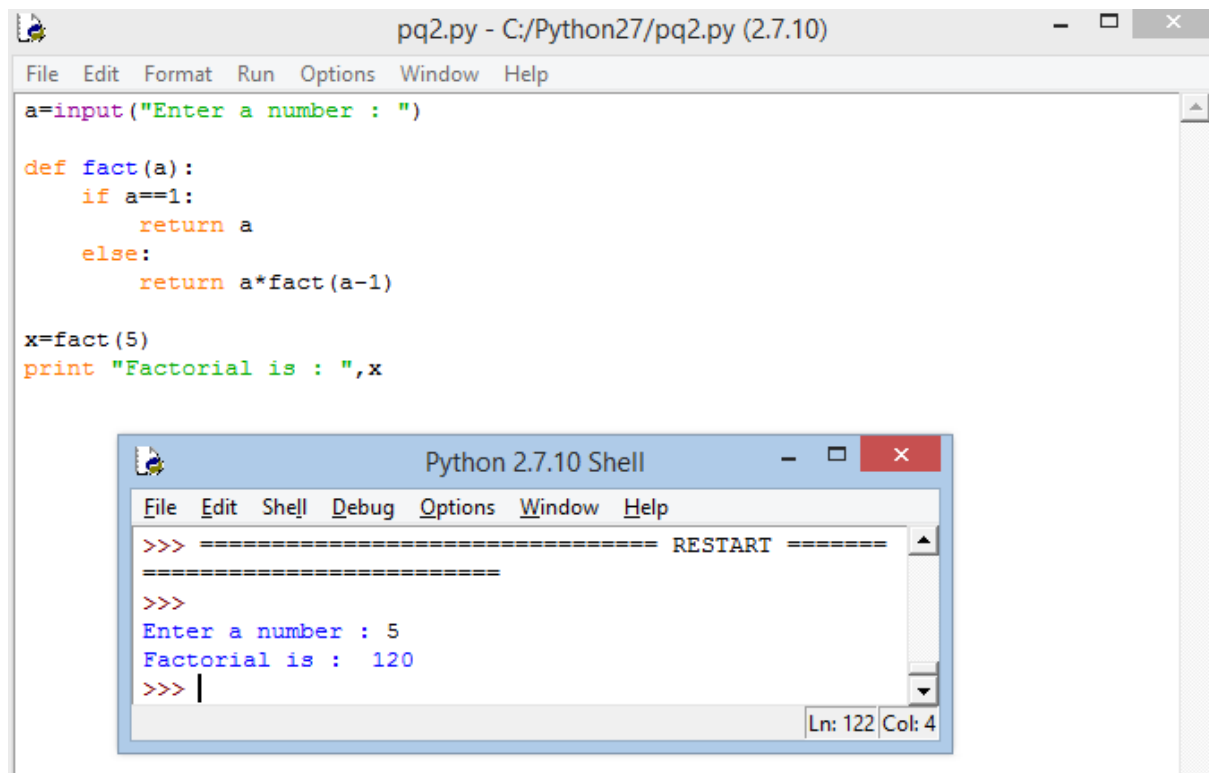


```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>> a
(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
>>> even=0
>>> odd=0
>>> for i in a:
    if i%2==0:
        even=even+1
    else:
        odd=odd+1

>>> "Total even number : ",even
('Total even number : ', 6)
>>> print "Total odd numbers : ",odd
Total odd numbers : 6
>>> |
```

Ln: 141 Col: 4

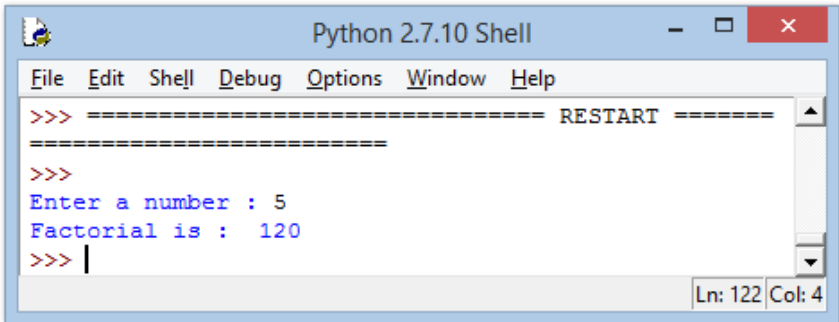
/\* Ques43. Program to find factorial of a number using function. \*/



```
pq2.py - C:/Python27/pq2.py (2.7.10)
File Edit Format Run Options Window Help
a=input("Enter a number : ")

def fact(a):
    if a==1:
        return a
    else:
        return a*fact(a-1)

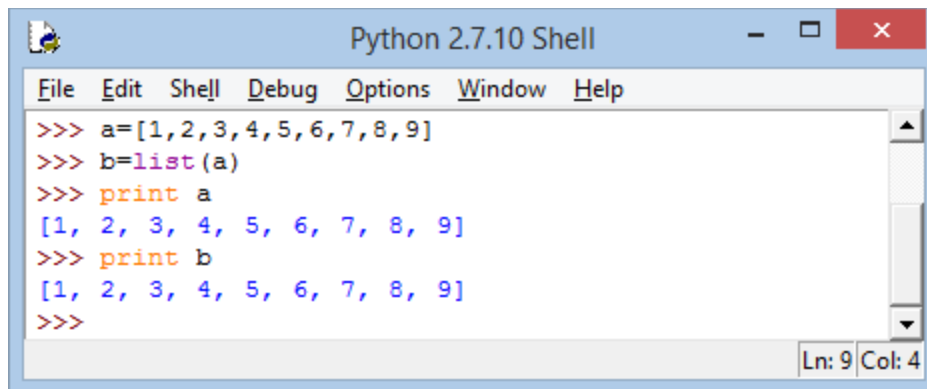
x=fact(5)
print "Factorial is : ",x
```

```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>> ===== RESTART =====
>>>
Enter a number : 5
Factorial is : 120
>>> |
```

Ln: 122 Col: 4

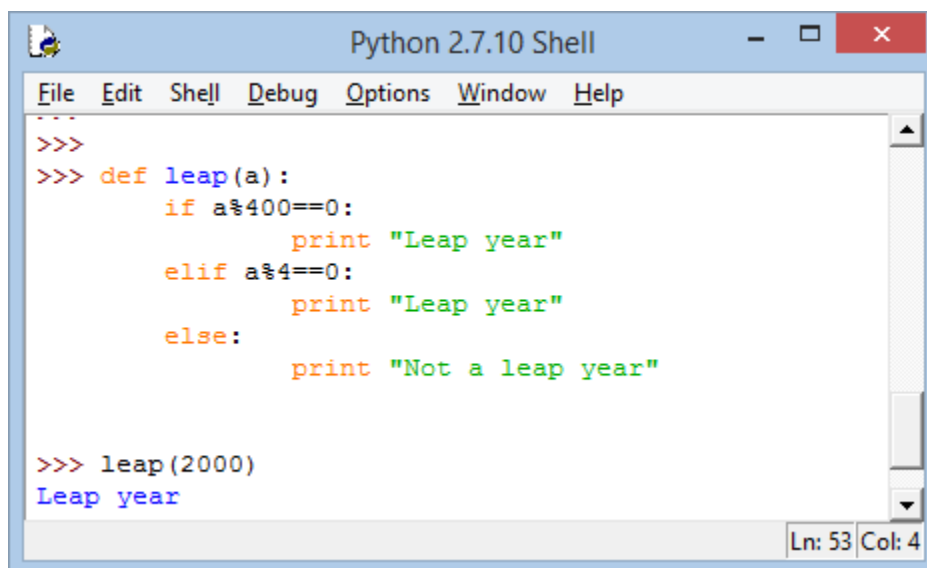
/\* Ques44. Program to copy one list into another list. \*/



```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>> a=[1,2,3,4,5,6,7,8,9]
>>> b=list(a)
>>> print a
[1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> print b
[1, 2, 3, 4, 5, 6, 7, 8, 9]
>>>
```

Ln: 9 Col: 4

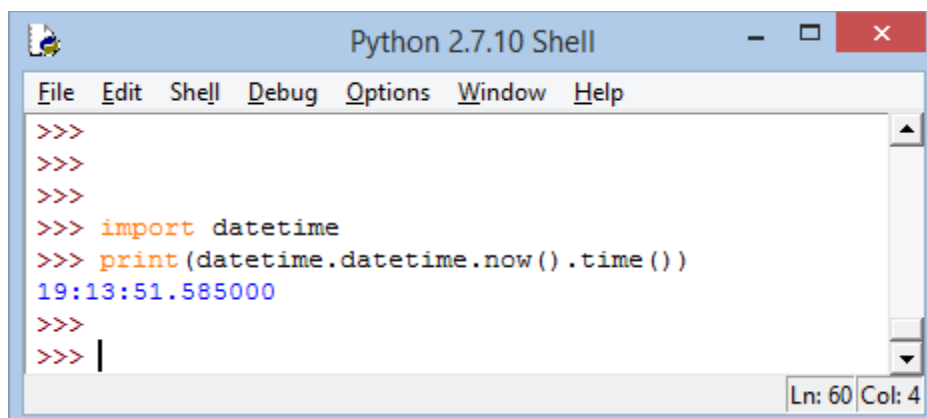
/\* Ques45. Program to find leap year or not. \*/



```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>>
>>> def leap(a):
>>>     if a%400==0:
>>>         print "Leap year"
>>>     elif a%4==0:
>>>         print "Leap year"
>>>     else:
>>>         print "Not a leap year"
>>>
>>> leap(2000)
Leap year
```

Ln: 53 Col: 4

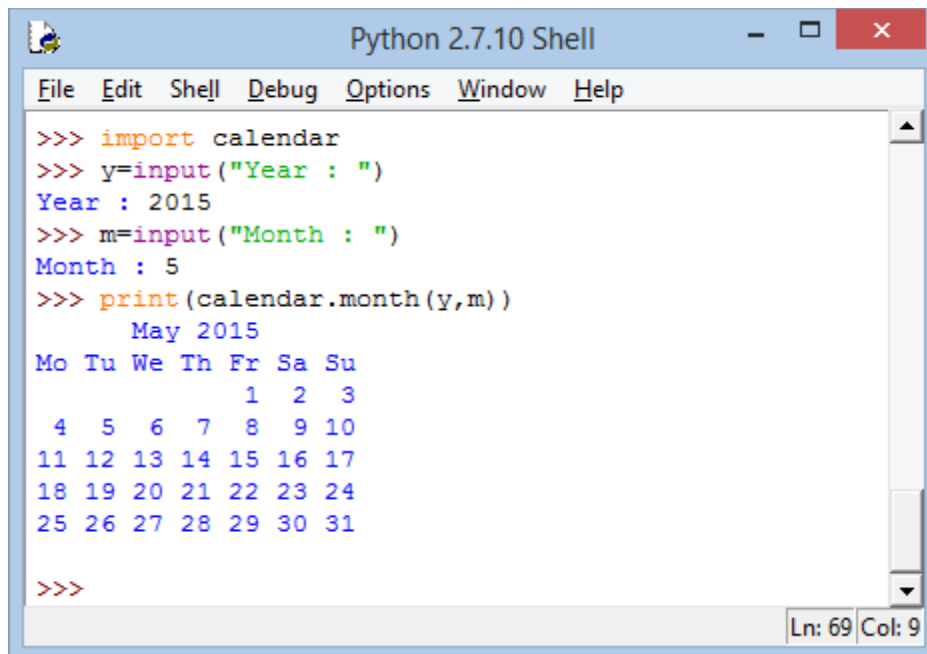
/\* Ques46. Program to print current time. \*/



```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>>
>>>
>>>
>>> import datetime
>>> print(datetime.datetime.now().time())
19:13:51.585000
>>>
>>> |
```

Ln: 60 Col: 4

/\* Ques47. Program to print calendar. \*/

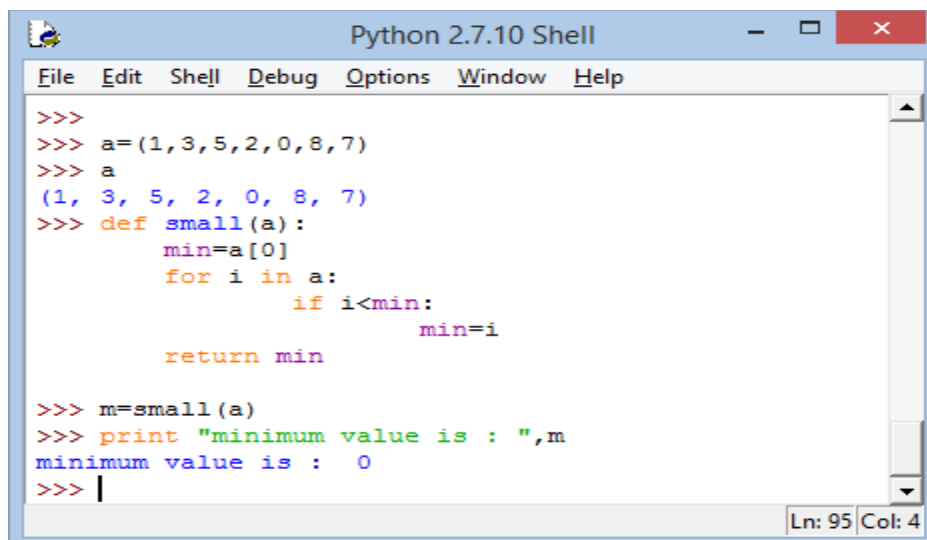


```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>> import calendar
>>> y=input("Year : ")
Year : 2015
>>> m=input("Month : ")
Month : 5
>>> print(calendar.month(y,m))
    May 2015
Mo Tu We Th Fr Sa Su
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

>>>
```

Ln: 69 Col: 9

/\* Ques48. Program to find minimum number in a given range of number. \*/

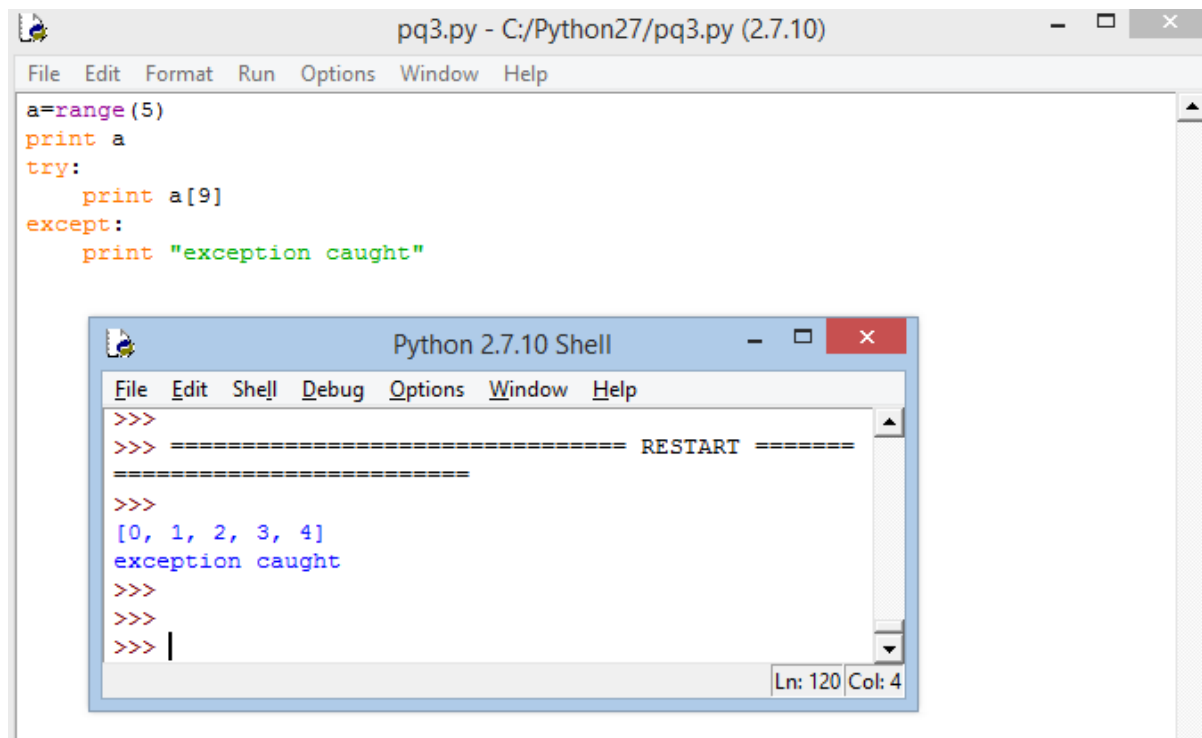


```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>>
>>> a=(1,3,5,2,0,8,7)
>>> a
(1, 3, 5, 2, 0, 8, 7)
>>> def small(a):
    min=a[0]
    for i in a:
        if i<min:
            min=i
    return min

>>> m=small(a)
>>> print "minimum value is : ",m
minimum value is :  0
>>> |
```

Ln: 95 Col: 4

/\* Ques49. Program to demonstrate exception handling. \*/



The image shows a Python 2.7.10 IDE window titled 'pq3.py - C:/Python27/pq3.py (2.7.10)'. The code in the editor is:

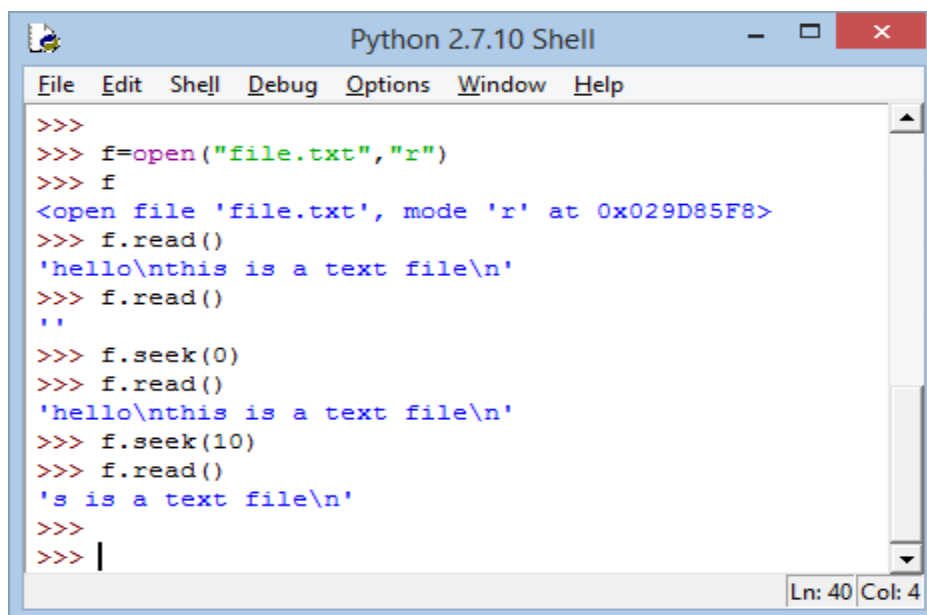
```
a=range(5)
print a
try:
    print a[9]
except:
    print "exception caught"
```

Below the editor is a 'Python 2.7.10 Shell' window. It shows the execution of the script, including a 'RESTART' message and the output of the code:

```
>>>
>>> ===== RESTART =====
>>>
[0, 1, 2, 3, 4]
exception caught
>>>
>>>
>>> |
```

The shell window status bar shows 'Ln: 120 Col: 4'.

/\* Ques50. Program to read the file and show the use of seek() function. \*/



The image shows a 'Python 2.7.10 Shell' window with the following code and output:

```
>>>
>>> f=open("file.txt","r")
>>> f
<open file 'file.txt', mode 'r' at 0x029D85F8>
>>> f.read()
'hello\nthis is a text file\n'
>>> f.read()
''
>>> f.seek(0)
>>> f.read()
'hello\nthis is a text file\n'
>>> f.seek(10)
>>> f.read()
's is a text file\n'
>>>
>>> |
```

The shell window status bar shows 'Ln: 40 Col: 4'.

**Written By**

Jyoti Goel  
(MCA, JIMS-Delhi)  
[jg.jyotigoel@gmail.com](mailto:jg.jyotigoel@gmail.com)

**Verified By**

Vikas Kumar Sharma  
(Corporate Python Trainer)  
[vikas.pypy@gmail.com](mailto:vikas.pypy@gmail.com)