

# Practical File

## Computer Science

### (Python)

**Name:**

**Class:**

**School:**

**Roll No:**

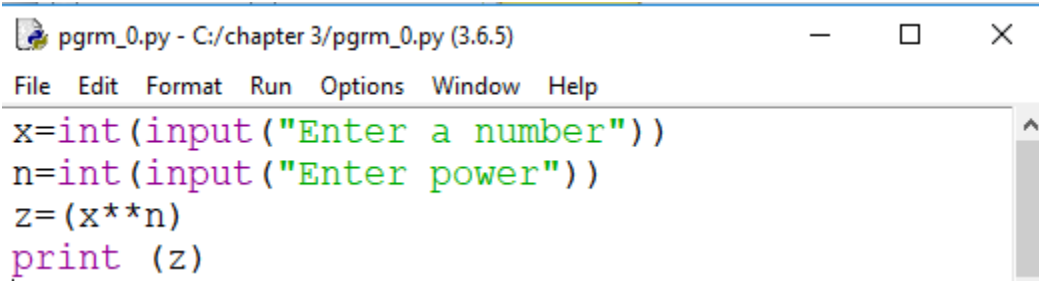
# Index

No.	Name of Practical	Date	Page no.	Sign of the Teacher
1.	WAP to compute $x^n$ of given two integers $x$ and $n$ .			
2.	WAP for calculating simple interest.			
3.	WAP to accept a number from the user and display whether it is an even number or odd number.			
4.	WAP to accept percentage of a student and display its grade accordingly.			
5.	WAP to print Fibonacci series up to certain limit.			
6.	WAP to display prime numbers up to a certain limit.			
7.	WAP to accept a number, find and display whether it's a Armstrong number or not.			
8.	WAP to accept a number and find out whether it is a perfect number or not.			
9.	WAP to print the sum of the series $1+x^1/1!+x^2/2!+.....x^n/(n)!$ - exponential series.			
10.	WAP to print the following pattern: 1 1 2 1 2 3			
11.	WAP to accept a string and display whether it is a palindrome.			
12.	WAP that counts the number of alphabets and digits, uppercase letters, lowercase letter, spaces and other characters in the string entered.			
13.	WAP to accept a string ( a sentence) and returns a string having first letter of each word in capital letter.			
14.	WAP to remove all odd numbers from the given			

	list.			
15.	WAP to display second largest element of a given list.			
16.	WAP to display cumulative elements of a given list.			
17.	WAP to display frequencies of all the elements of a list.			
18.	WAP in Python to display those strings which are string with 'A' of given list.			
19.	WAP in Python to find and display the sum of all the values which are ending with 3 from a list.			
20.	WAP to shift the negative number to left and the positive numbers to right.			
21.	WAP to swap the content with next value divisible by 7.			
22.	WAP to accept values from user and create a tuple.			
23.	Write a program to input total number of sections and stream name in 11 <sup>th</sup> class and display all information on the output screen.			
24.	Write a Python program to input names of 'n' countries and their capital and currency, store it in a dictionary and display in tabular form. Also search and display for a particular country.			
	<b>SQL</b>			
	25 Queries			

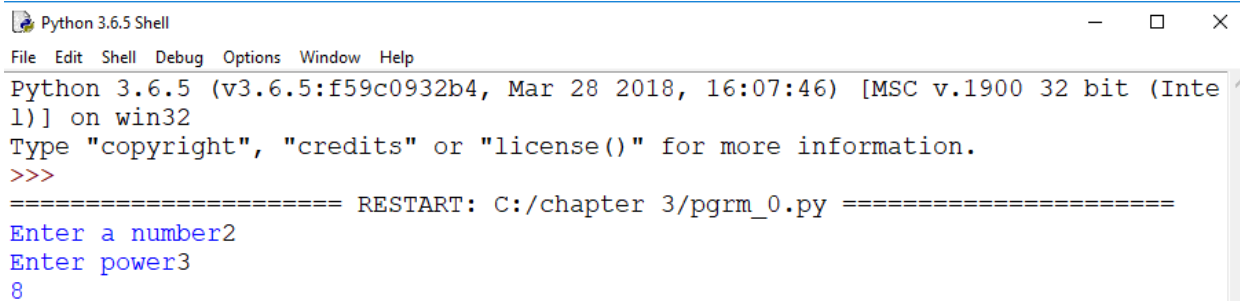
## Program -1 WAP to compute $x^n$ of given two integers x and n.

### Code:



```
pgrm_0.py - C:/chapter 3/pgrm_0.py (3.6.5)
File Edit Format Run Options Window Help
x=int(input("Enter a number"))
n=int(input("Enter power"))
z=(x**n)
print(z)
```

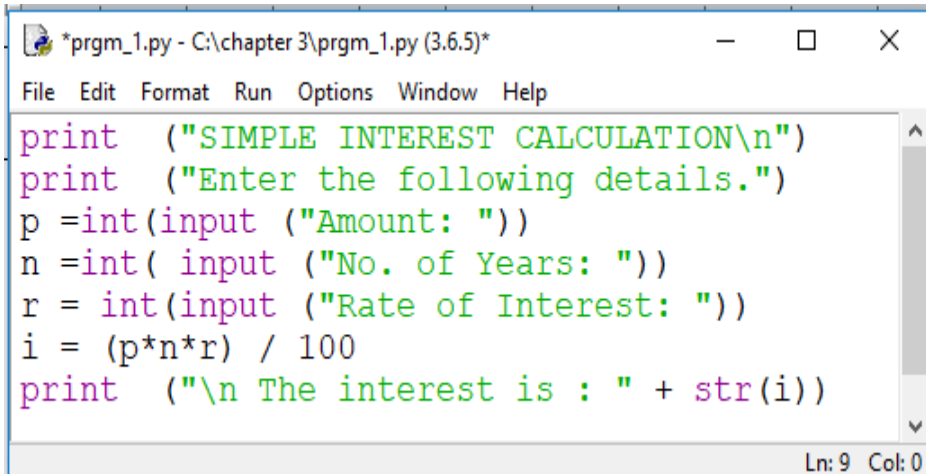
### \*\*\*\*\*Output of the program\*\*\*\*\*



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/chapter 3/pgrm_0.py =====
Enter a number2
Enter power3
8
```

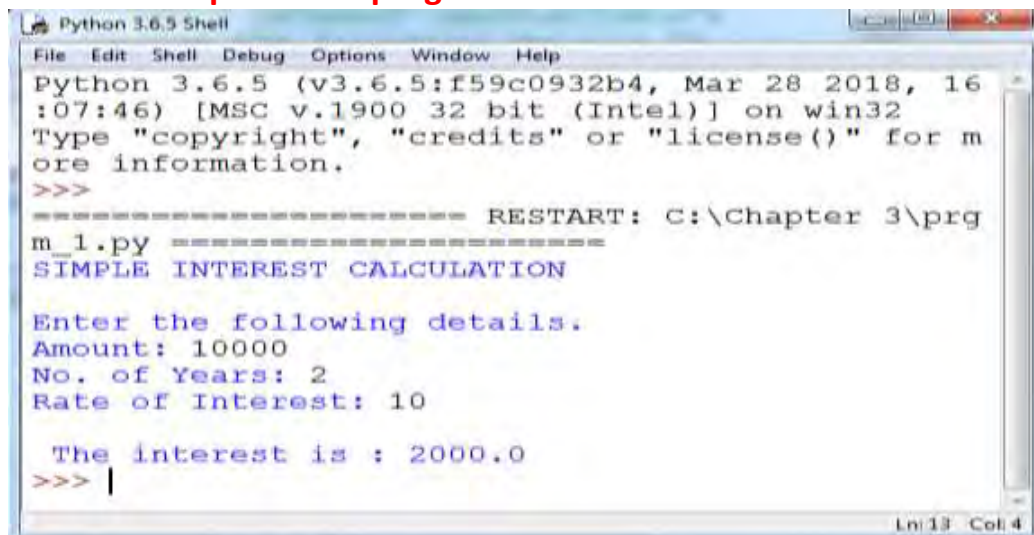
## Program 2: WAP for calculating simple interest.

### Code:



```
*prgm_1.py - C:\chapter 3\prgm_1.py (3.6.5)*
File Edit Format Run Options Window Help
print ("SIMPLE INTEREST CALCULATION\n")
print ("Enter the following details.")
p =int(input ("Amount: "))
n =int( input ("No. of Years: "))
r = int(input ("Rate of Interest: "))
i = (p*n*r) / 100
print ("\n The interest is : " + str(i))
Ln: 9 Col: 0
```

### \*\*\*\*\*Output of the program\*\*\*\*\*

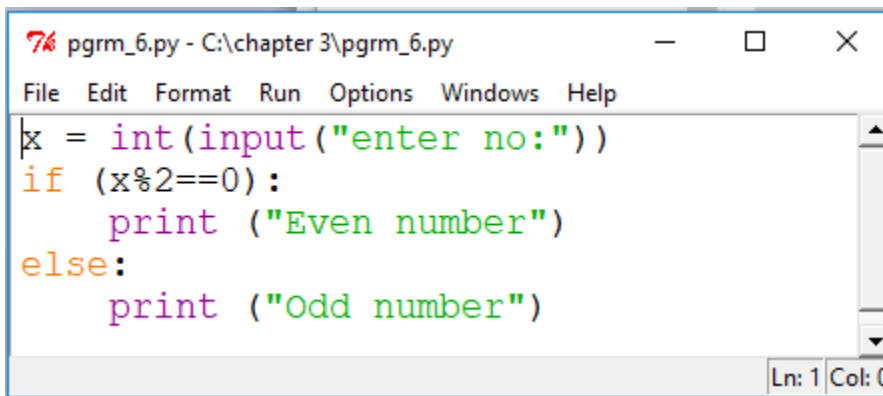


```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
----- RESTART: C:\Chapter 3\prgm_1.py -----
SIMPLE INTEREST CALCULATION
Enter the following details.
Amount: 10000
No. of Years: 2
Rate of Interest: 10

The interest is : 2000.0
>>> |
Ln: 13 Col: 4
```

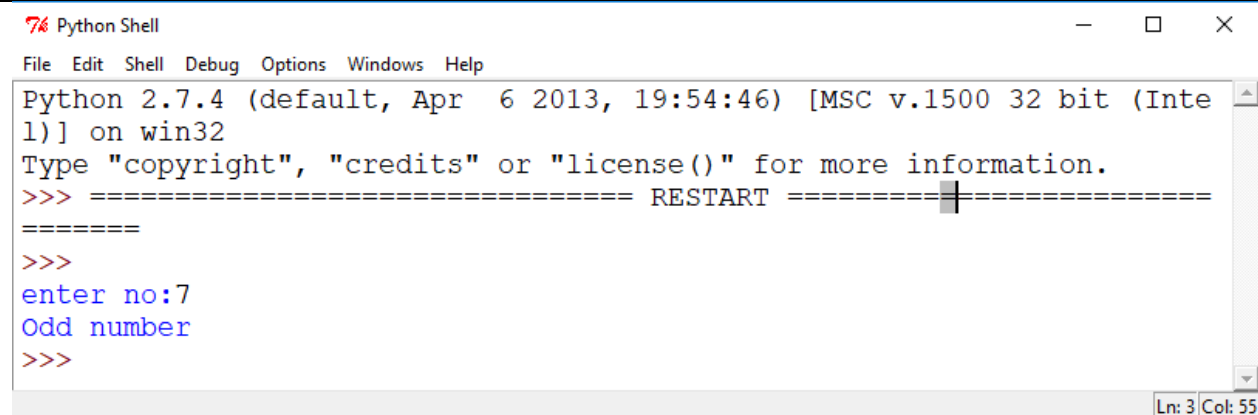
**Program 3: WAP to accept a number from the user and display whether it is an even number or odd number.**

**Code:**



```
pgrm_6.py - C:\chapter 3\pgrm_6.py
File Edit Format Run Options Windows Help
x = int(input("enter no:"))
if (x%2==0):
    print ("Even number")
else:
    print ("Odd number")
Ln: 1 Col: 0
```

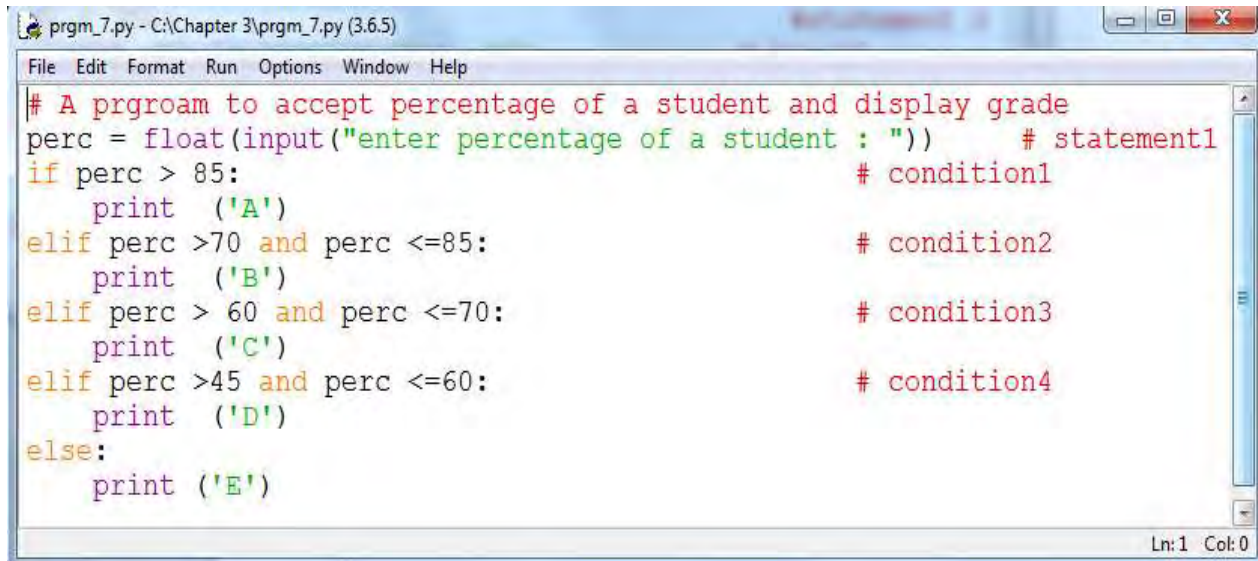
**\*\*\*\*\*Output of the program\*\*\*\*\***



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 2.7.4 (default, Apr 6 2013, 19:54:46) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
>>> enter no:7
>>> Odd number
>>>
Ln: 3 Col: 55
```

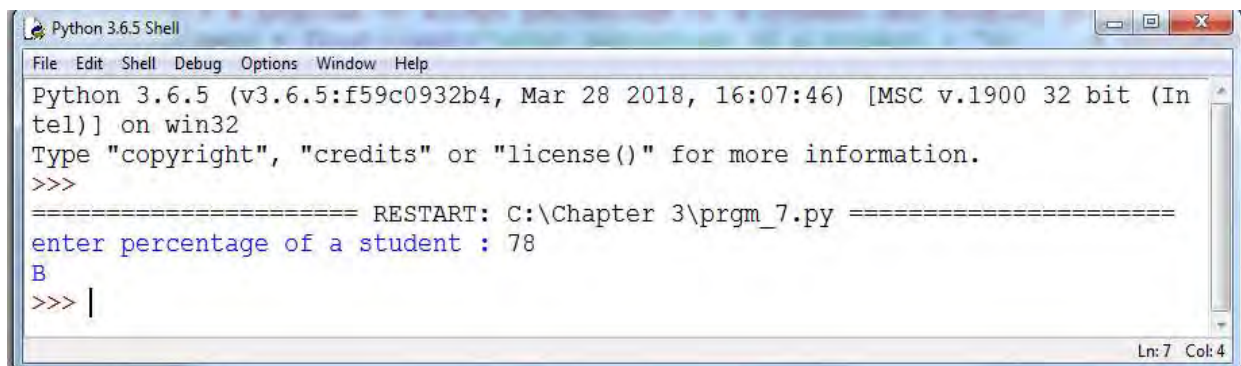
## Program 4: WAP to accept percentage of a student and display its grade accordingly.

### Code:



```
prgm_7.py - C:\Chapter 3\prgm_7.py (3.6.5)
File Edit Format Run Options Window Help
# A prgroam to accept percentage of a student and display grade
perc = float(input("enter percentage of a student : "))      # statement1
if perc > 85:                                               # condition1
    print ('A')
elif perc >70 and perc <=85:                               # condition2
    print ('B')
elif perc > 60 and perc <=70:                             # condition3
    print ('C')
elif perc >45 and perc <=60:                              # condition4
    print ('D')
else:
    print ('E')
Ln:1 Col:0
```

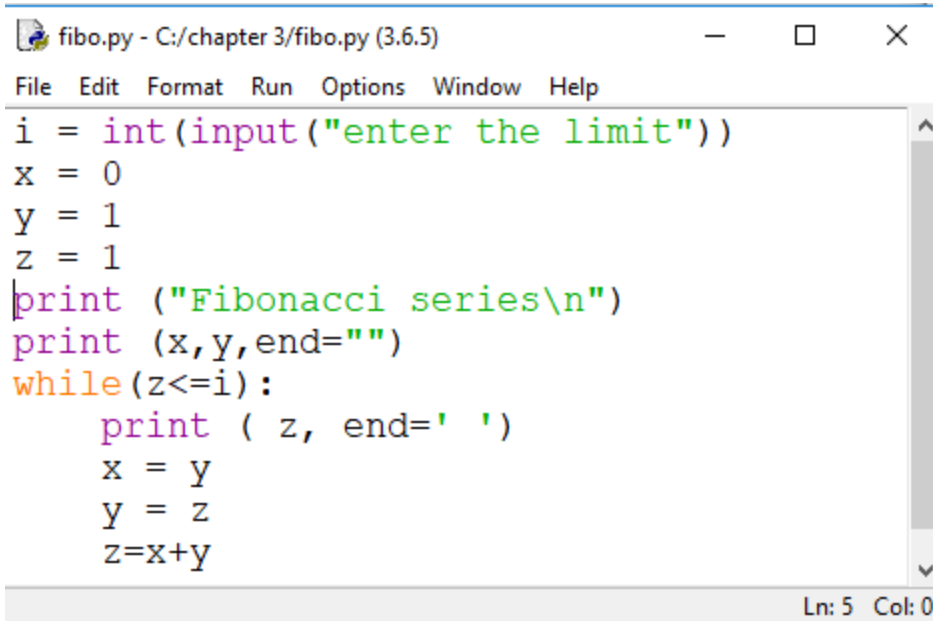
### \*\*\*\*\*Output of the program\*\*\*\*\*



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Chapter 3\prgm_7.py =====
enter percentage of a student : 78
B
>>> |
Ln:7 Col:4
```

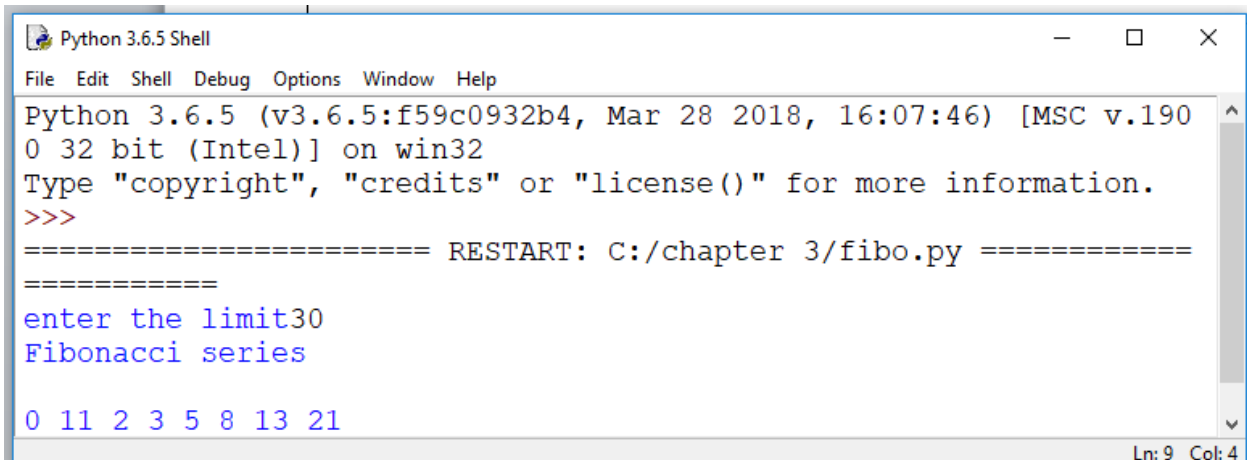
## Program 5: WAP to print Fibonacci series up to certain limit.

### Code:



```
fibonacci.py - C:/chapter 3/fibonacci.py (3.6.5)
File Edit Format Run Options Window Help
i = int(input("enter the limit"))
x = 0
y = 1
z = 1
print ("Fibonacci series\n")
print (x,y,end=" ")
while(z<=i):
    print ( z, end=' ')
    x = y
    y = z
    z=x+y
Ln: 5 Col: 0
```

### \*\*\*\*\*Output of the program\*\*\*\*\*

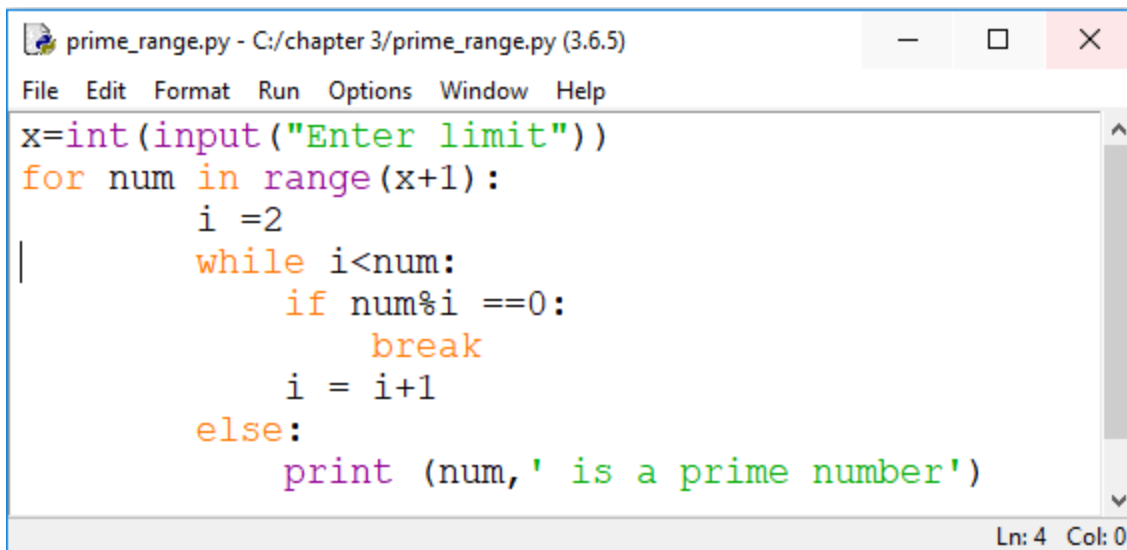


```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.190
0 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/chapter 3/fibonacci.py =====
=====
enter the limit30
Fibonacci series
0 1 1 2 3 5 8 13 21
Ln: 9 Col: 4
```



## Program 6: WAP to display prime numbers up to a certain limit.

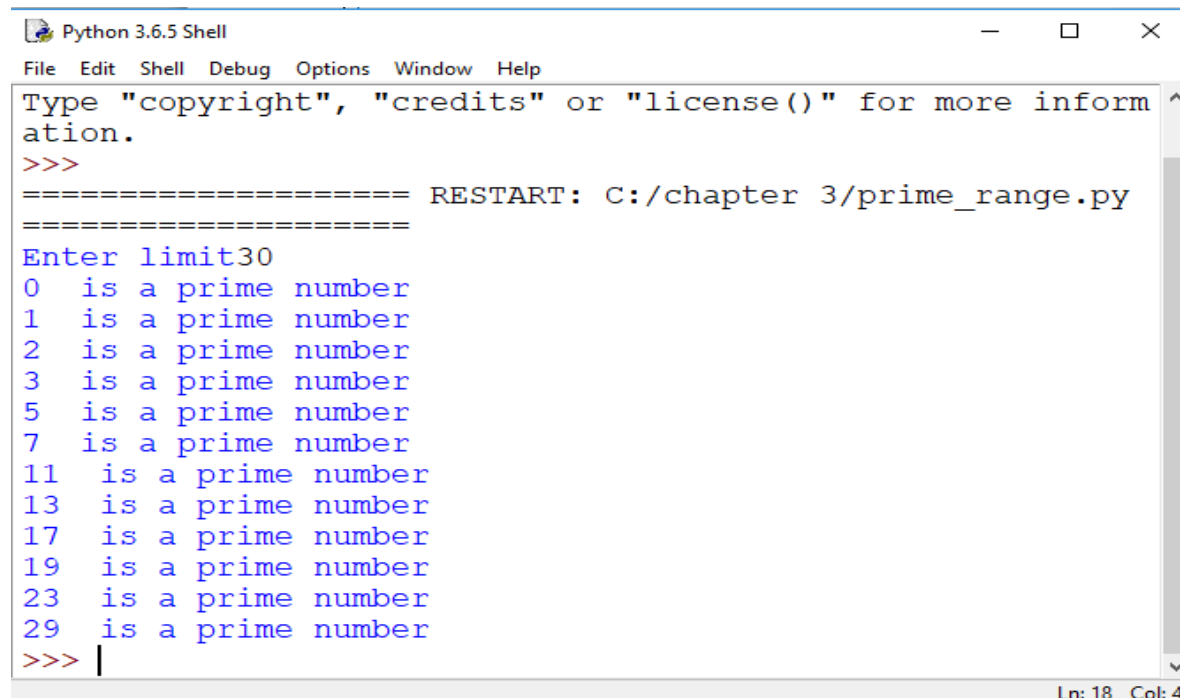
### Code:



```
prime_range.py - C:/chapter 3/prime_range.py (3.6.5)
File Edit Format Run Options Window Help
x=int(input("Enter limit"))
for num in range(x+1):
    i =2
    while i<num:
        if num%i ==0:
            break
        i = i+1
    else:
        print (num,' is a prime number')
```

Ln: 4 Col: 0

### \*\*\*\*\*Output of the program\*\*\*\*\*

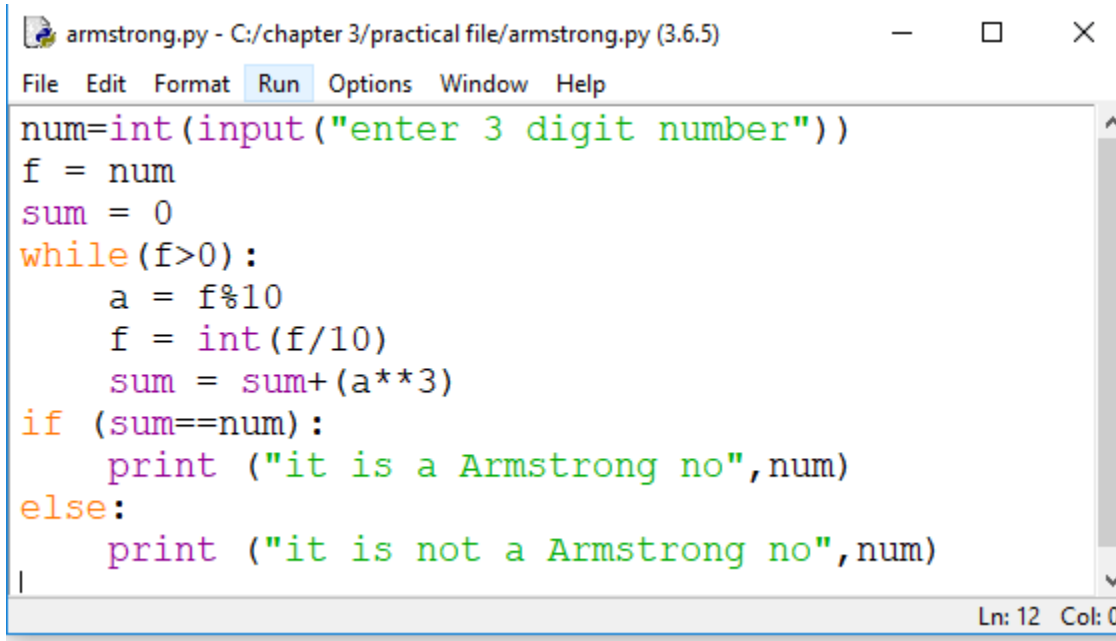


```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/chapter 3/prime_range.py
=====
Enter limit30
0 is a prime number
1 is a prime number
2 is a prime number
3 is a prime number
5 is a prime number
7 is a prime number
11 is a prime number
13 is a prime number
17 is a prime number
19 is a prime number
23 is a prime number
29 is a prime number
>>> |
```

Ln: 18 Col: 4

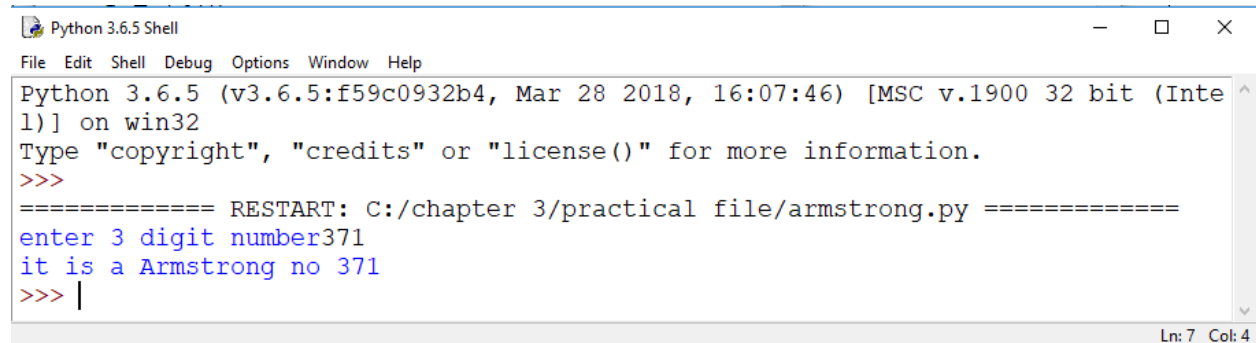
**Program 7: WAP to accept a number, find and display whether it's a Armstrong number or not.**

**Code:**



```
armstrong.py - C:/chapter 3/practical file/armstrong.py (3.6.5)
File Edit Format Run Options Window Help
num=int(input("enter 3 digit number"))
f = num
sum = 0
while(f>0):
    a = f%10
    f = int(f/10)
    sum = sum+(a**3)
if (sum==num):
    print ("it is a Armstrong no",num)
else:
    print ("it is not a Armstrong no",num)
Ln: 12 Col: 0
```

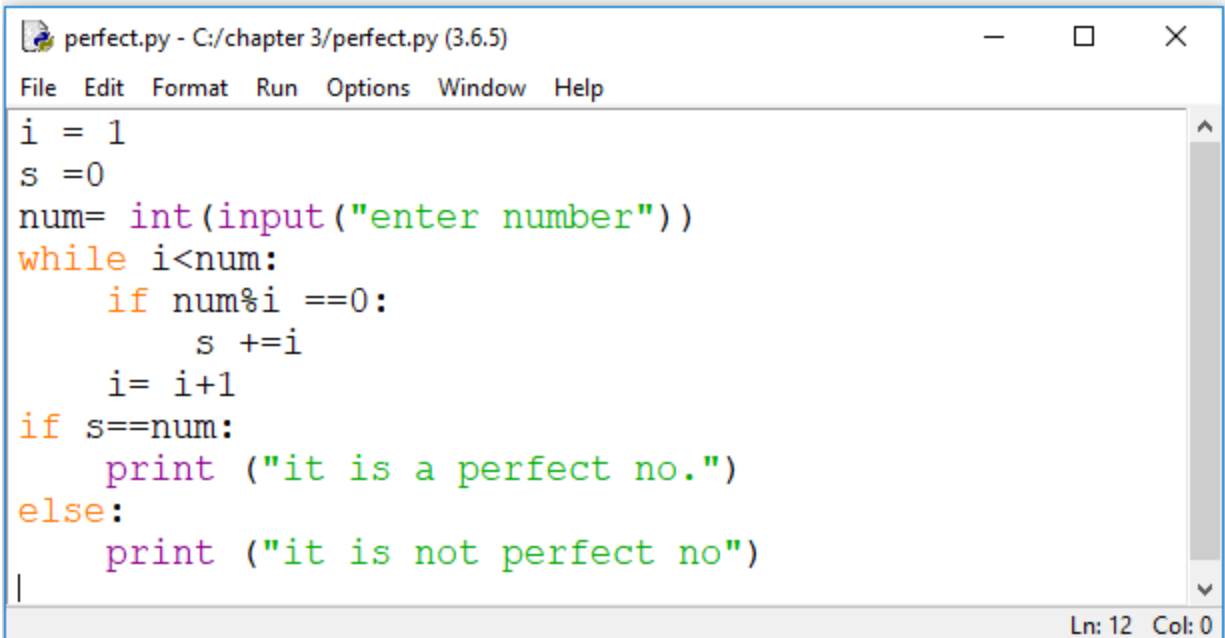
**\*\*\*\*\*Output of the program\*\*\*\*\***



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/chapter 3/practical file/armstrong.py =====
enter 3 digit number371
it is a Armstrong no 371
>>> |
Ln: 7 Col: 4
```

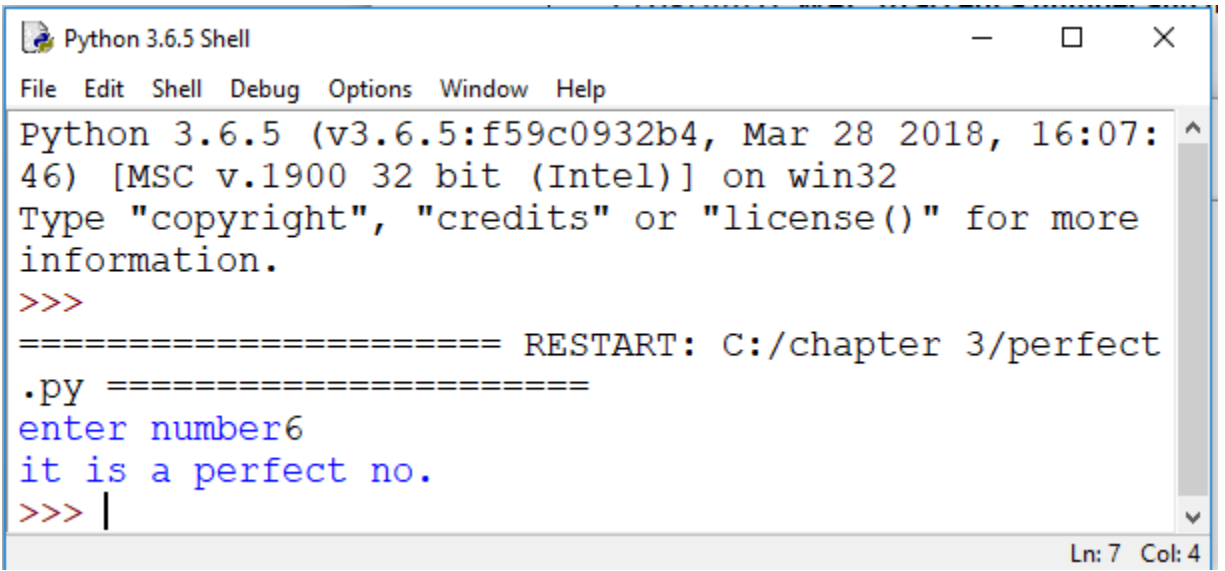
**Program 8: WAP to accept a number and find out whether it is a perfect number or not.**

**Code:**



```
perfect.py - C:/chapter 3/perfect.py (3.6.5)
File Edit Format Run Options Window Help
i = 1
s = 0
num= int(input("enter number"))
while i<num:
    if num%i ==0:
        s +=i
    i= i+1
if s==num:
    print ("it is a perfect no.")
else:
    print ("it is not perfect no")
|
Ln: 12 Col: 0
```

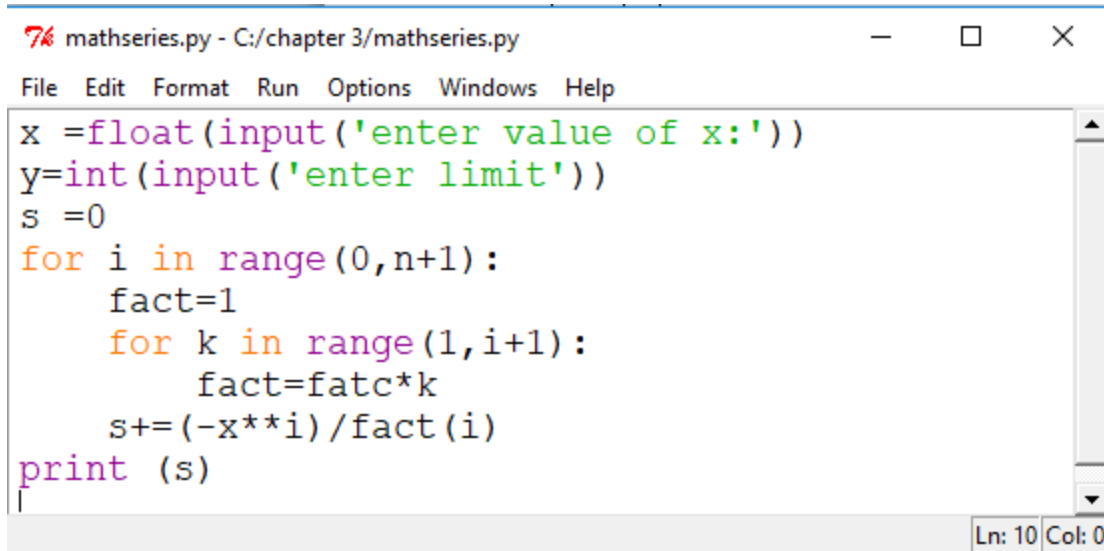
**\*\*\*\*\*Output of the program\*\*\*\*\***



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:
46) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more
information.
>>>
===== RESTART: C:/chapter 3/perfect
.py =====
enter number6
it is a perfect no.
>>> |
Ln: 7 Col: 4
```

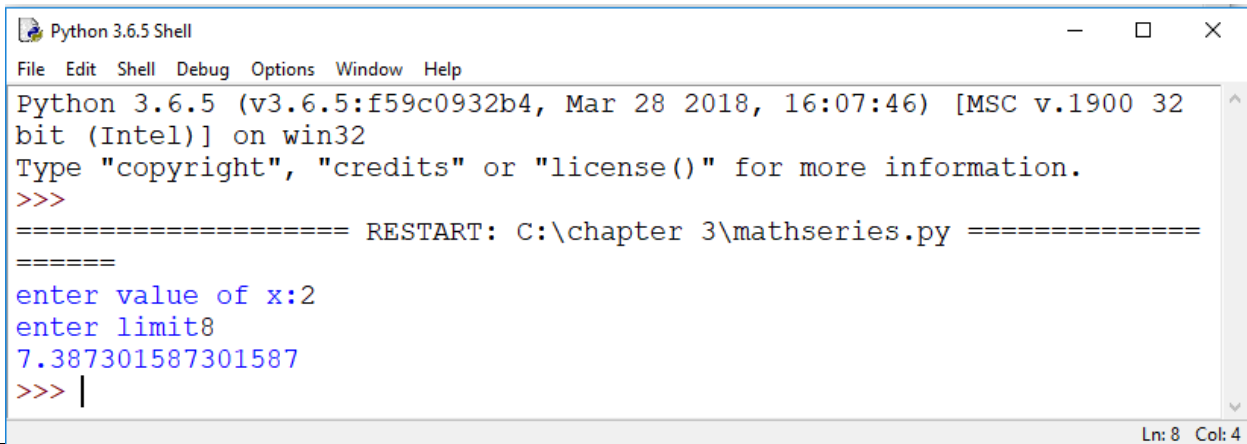
**Program 9: WAP to print the sum of the series  $1+x1/1!+x2/2!+.....xn/(n)!$ - exponential series.**

**Code:**



```
7% mathseries.py - C:/chapter 3/mathseries.py
File Edit Format Run Options Windows Help
x =float(input('enter value of x:'))
y=int(input('enter limit'))
s =0
for i in range(0,n+1):
    fact=1
    for k in range(1,i+1):
        fact=fatc*k
    s+=(-x**i)/fact(i)
print (s)
Ln: 10 Col: 0
```

**\*\*\*\*\*Output of the program\*\*\*\*\***

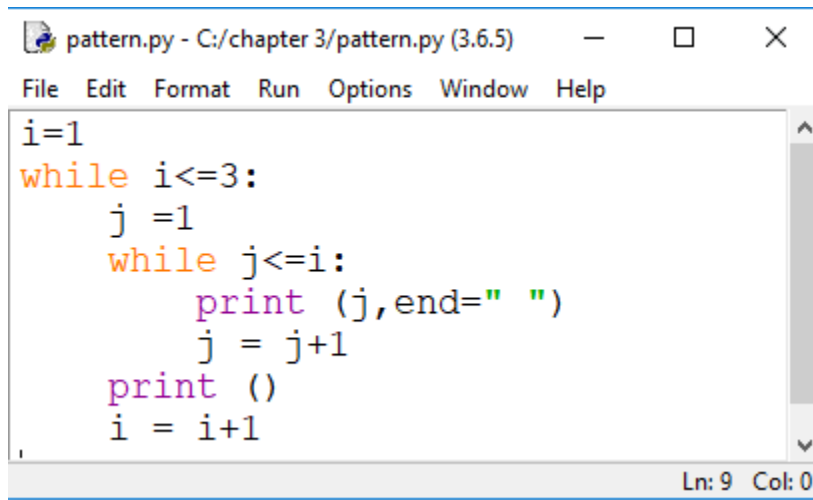


```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32
bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\chapter 3\mathseries.py =====
=====
enter value of x:2
enter limit8
7.387301587301587
>>> |
Ln: 8 Col: 4
```

**Program 10: WAP to print the following pattern:**

1  
1 2  
1 2 3

**Code:**



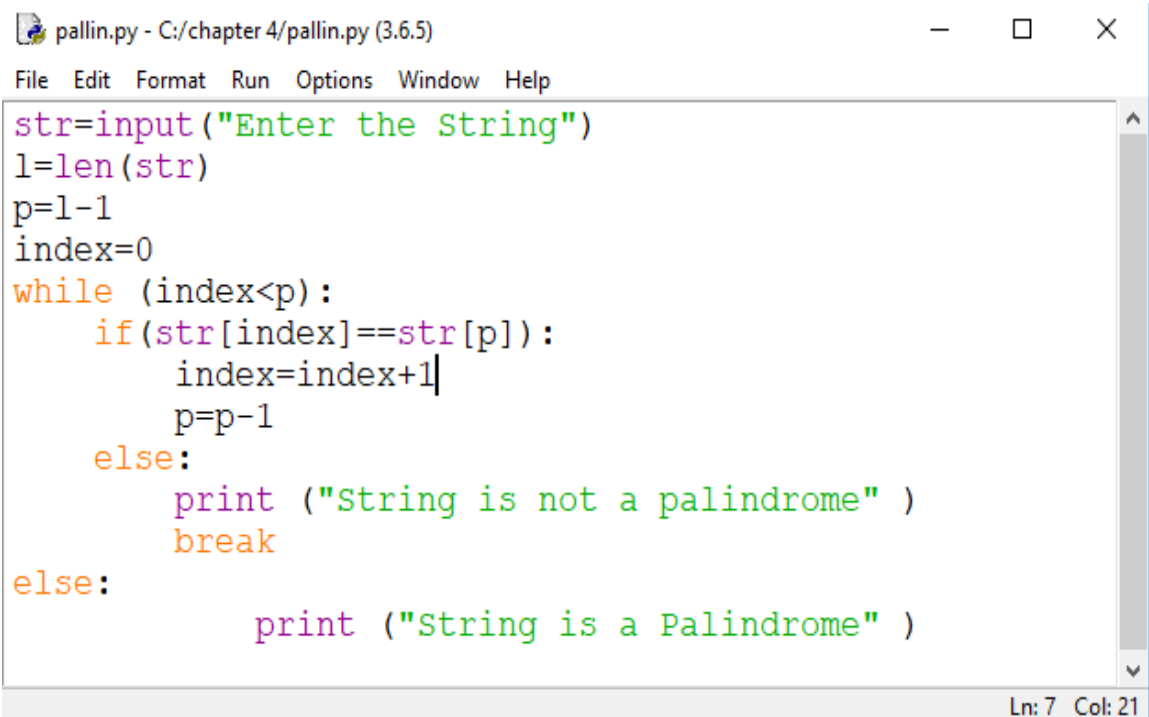
```
pattern.py - C:/chapter 3/pattern.py (3.6.5)
File Edit Format Run Options Window Help
i=1
while i<=3:
    j =1
    while j<=i:
        print (j,end=" ")
        j = j+1
    print ()
    i = i+1
Ln: 9 Col: 0
```

**\*\*\*\*\*Output of the program\*\*\*\*\***

1  
1 2  
1 2 3

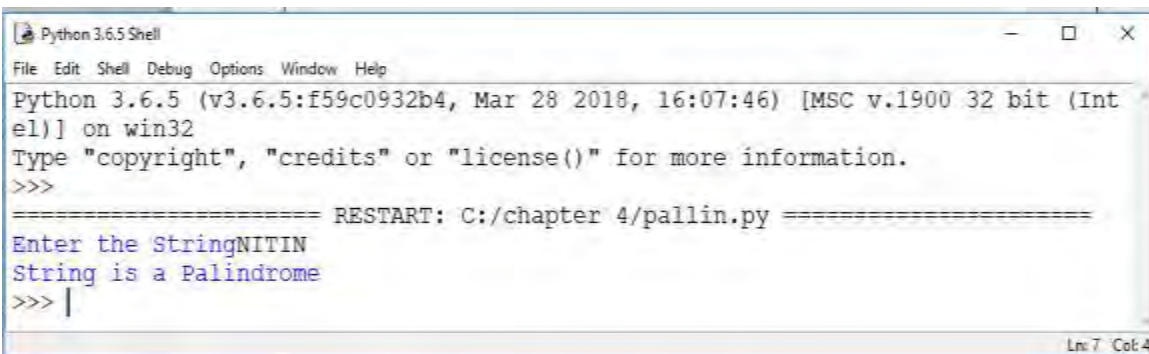
## Program 11: WAP to accept a string and display whether it is a palindrome.

### Code:



```
pallin.py - C:/chapter 4/pallin.py (3.6.5)
File Edit Format Run Options Window Help
str=input("Enter the String")
l=len(str)
p=l-1
index=0
while (index<p):
    if (str[index]==str[p]):
        index=index+1
        p=p-1
    else:
        print ("String is not a palindrome" )
        break
else:
    print ("String is a Palindrome" )
Ln: 7 Col: 21
```

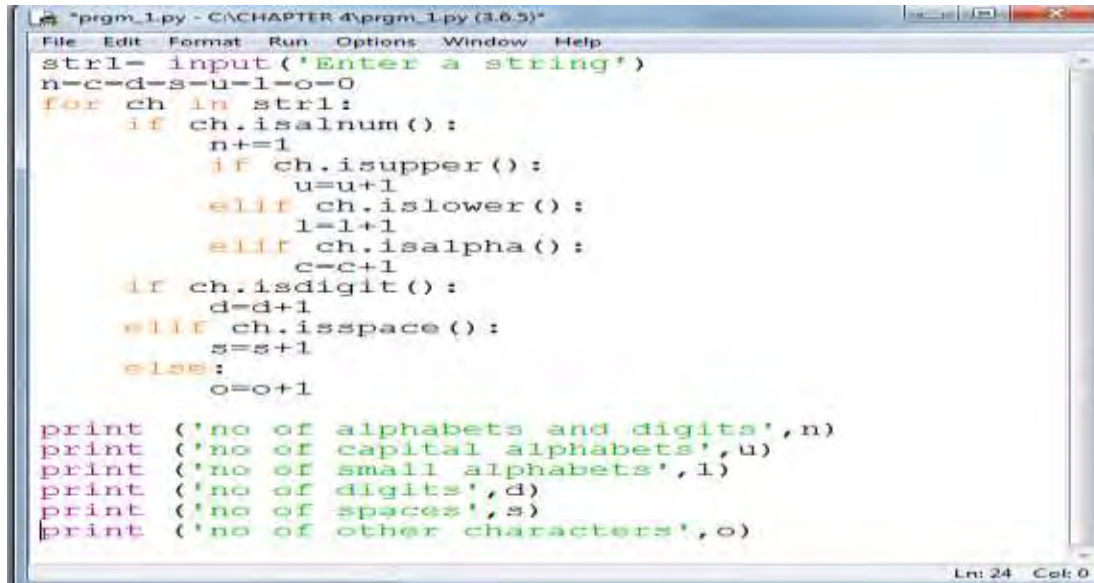
### \*\*\*\*\*Output of the program\*\*\*\*\*



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Int
el)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/chapter 4/pallin.py =====
Enter the StringNITIN
String is a Palindrome
>>> |
Ln: 7 Col: 4
```

**Program 12-WAP that counts the number of alphabets and digits, uppercase letters, lowercase letter, spaces and other characters in the string entered.**

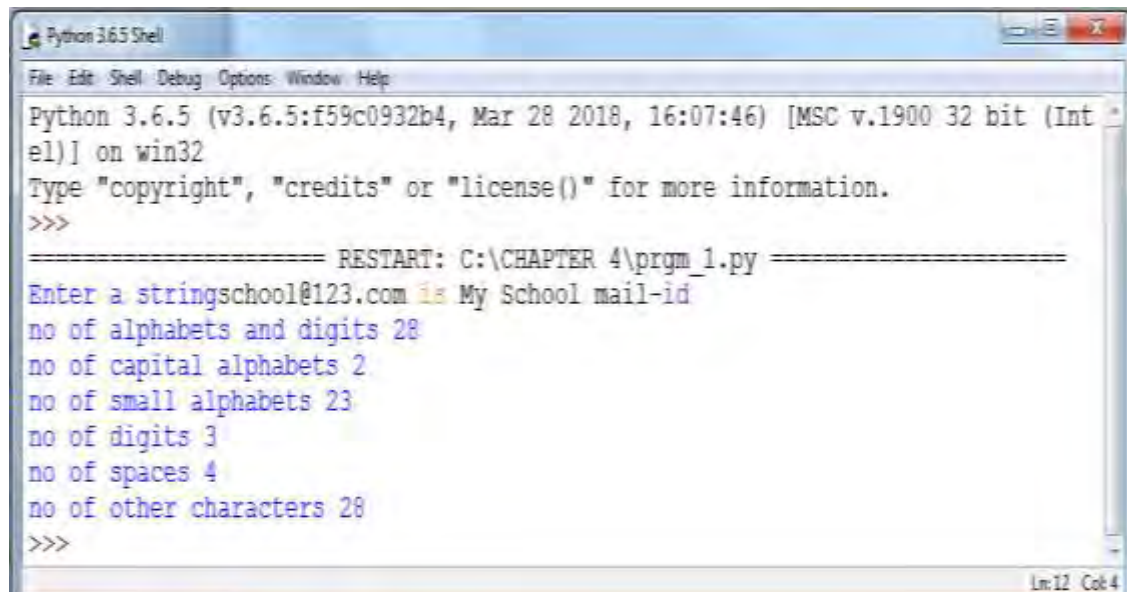
**Code:**



```
prgm_1.py - C:\CHAPTER 4\prgm_1.py (3.6.5)
File Edit Format Run Options Window Help
str1= input('Enter a string')
n=c=d=s=u=l=o=0
for ch in str1:
    if ch.isalnum():
        n+=1
        if ch.isupper():
            u=u+1
        elif ch.islower():
            l=l+1
        elif ch.isalpha():
            c=c+1
    if ch.isdigit():
        d=d+1
    elif ch.isspace():
        s=s+1
    else:
        o=o+1

print ('no of alphabets and digits',n)
print ('no of capital alphabets',u)
print ('no of small alphabets',l)
print ('no of digits',d)
print ('no of spaces',s)
print ('no of other characters',o)
```

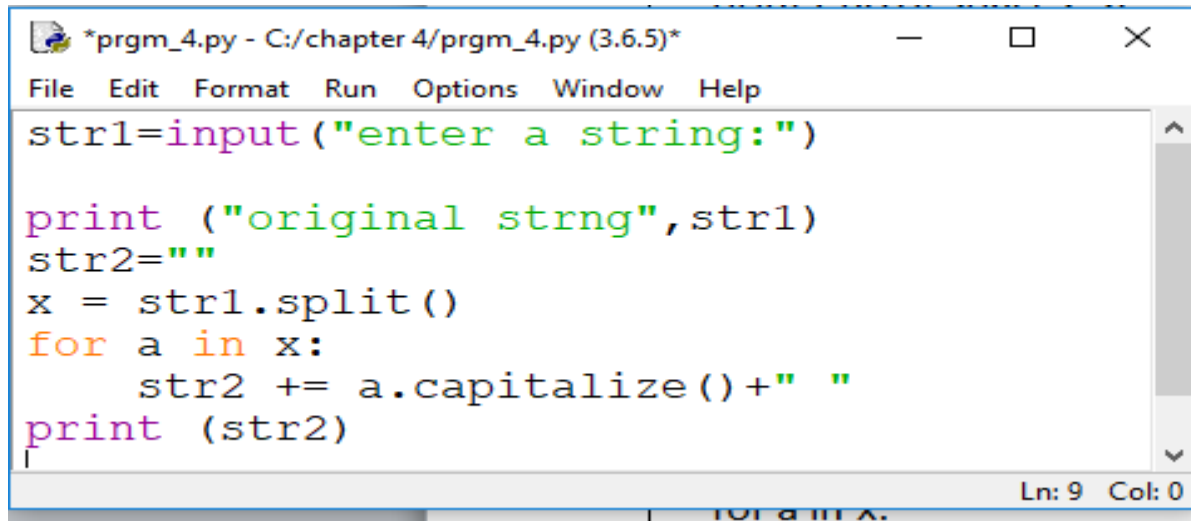
**\*\*\*\*\*Output of the program\*\*\*\*\***



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Int
el)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
----- RESTART: C:\CHAPTER 4\prgm_1.py -----
Enter a stringschool@123.com is My School mail-id
no of alphabets and digits 28
no of capital alphabets 2
no of small alphabets 23
no of digits 3
no of spaces 4
no of other characters 28
>>>
```

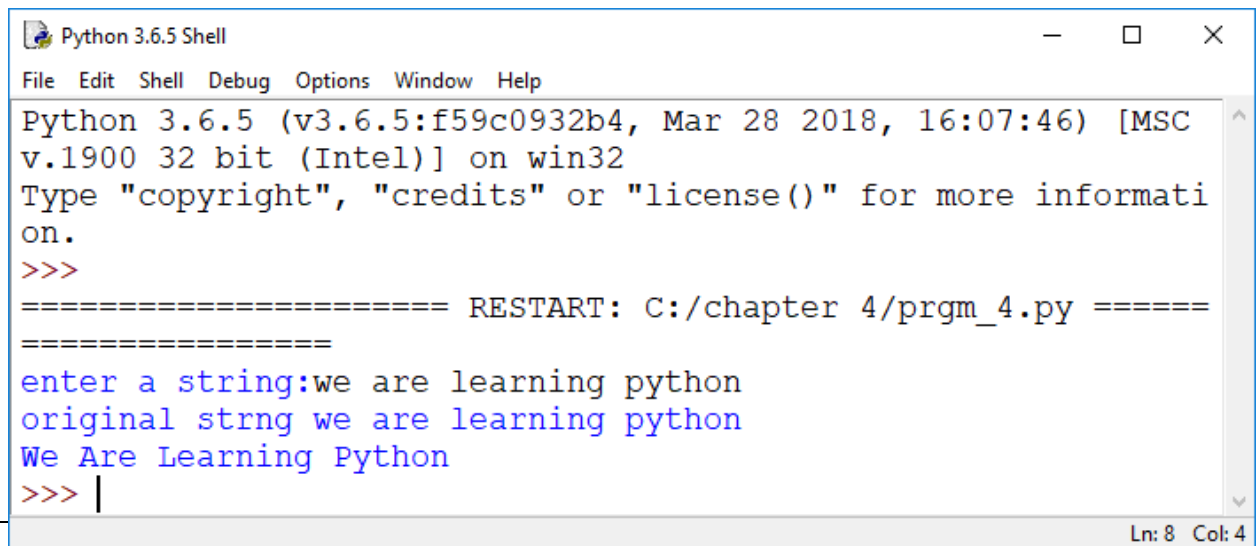
**Program 13 WAP to accept a string ( a sentence) and returns a string having first letter of each word in capital letter.**

**Code:**



```
*prgm_4.py - C:/chapter 4/prgm_4.py (3.6.5)*
File Edit Format Run Options Window Help
str1=input("enter a string:")
print ("original strng",str1)
str2=""
x = str1.split()
for a in x:
    str2 += a.capitalize()+" "
print (str2)
Ln: 9 Col: 0
```

**\*\*\*\*\*Output of the program\*\*\*\*\***

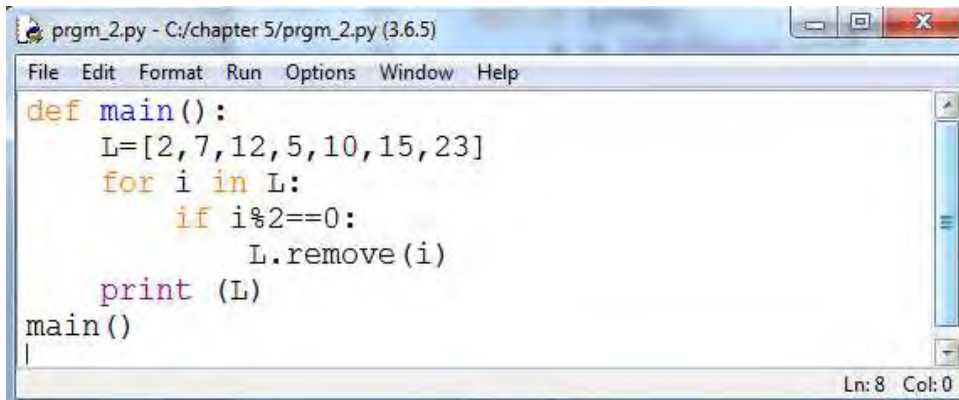


```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC
v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more informati
on.
>>>
===== RESTART: C:/chapter 4/prgm_4.py =====
enter a string:we are learning python
original strng we are learning python
We Are Learning Python
>>> |
Ln: 8 Col: 4
```



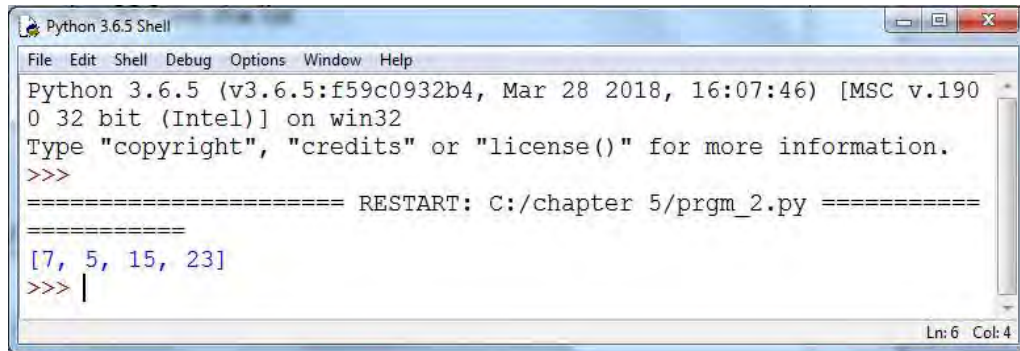
## Program 14 -WAP to remove all odd numbers from the given list.

**Code:**



```
prgm_2.py - C:/chapter 5/prgm_2.py (3.6.5)
File Edit Format Run Options Window Help
def main():
    L=[2,7,12,5,10,15,23]
    for i in L:
        if i%2==0:
            L.remove(i)
    print (L)
main()
Ln: 8 Col: 0
```

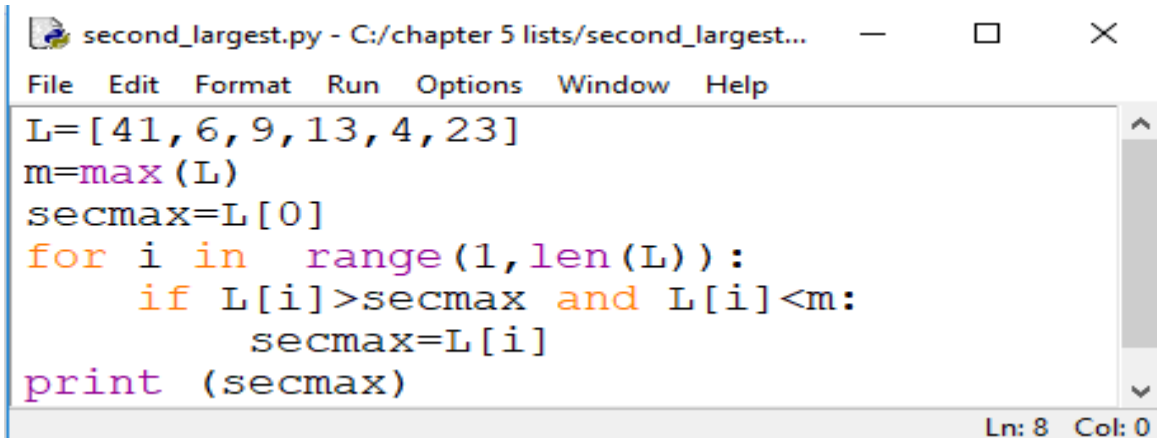
**\*\*\*\*\*Output of the program\*\*\*\*\***



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.190
0 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/chapter 5/prgm_2.py =====
[7, 5, 15, 23]
>>> |
Ln: 6 Col: 4
```

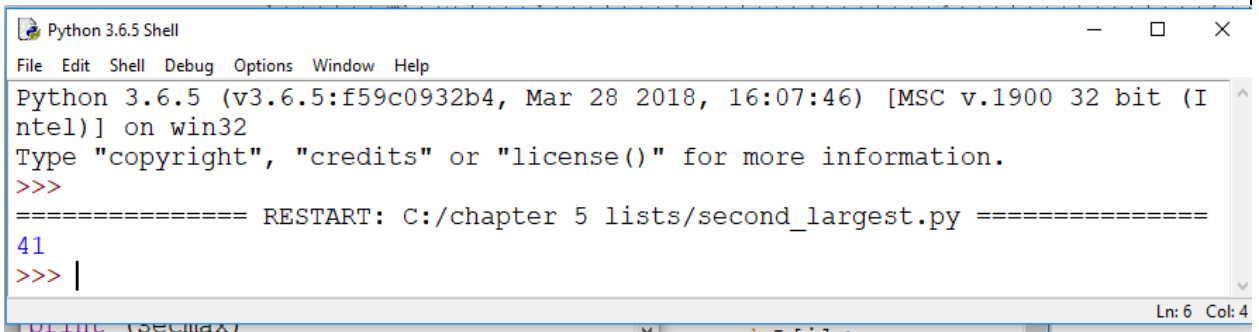
## Program 15 WAP to display second largest element of a given list.

### Code:



```
second_largest.py - C:/chapter 5 lists/second_largest...  -  □  ×
File Edit Format Run Options Window Help
L=[41, 6, 9, 13, 4, 23]
m=max(L)
secmax=L[0]
for i in range(1, len(L)):
    if L[i]>secmax and L[i]<m:
        secmax=L[i]
print (secmax)
Ln: 8 Col: 0
```

### \*\*\*\*\*Output of the program\*\*\*\*\*



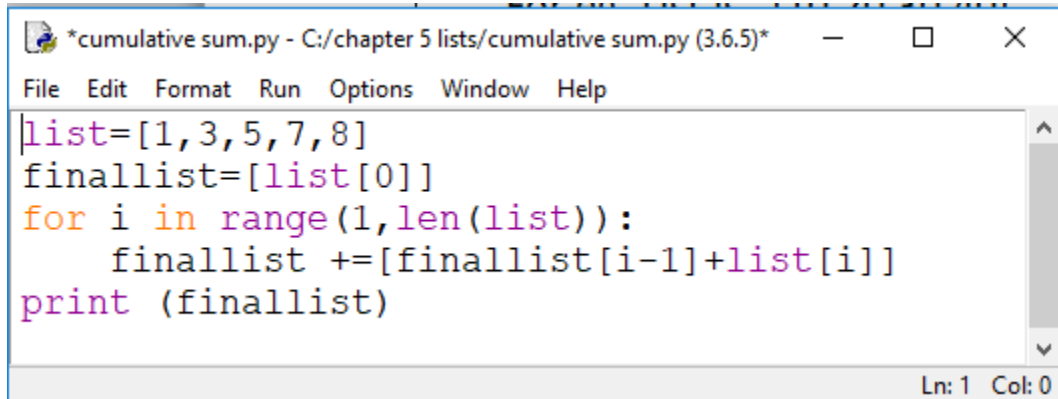
```
Python 3.6.5 Shell  -  □  ×
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (I
ntel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/chapter 5 lists/second_largest.py =====
41
>>> |
Ln: 6 Col: 4
```

**Program 16 WAP to display cumulative elements of a given list.**

For eg. List is [10,20,30,40]

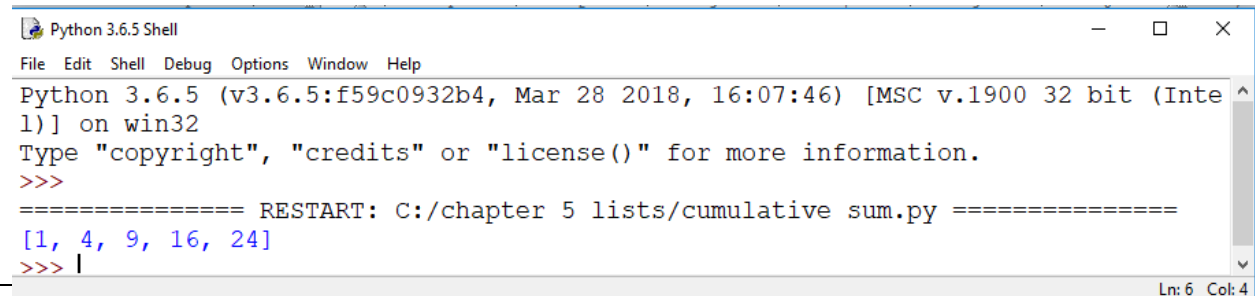
output should be [10, 30, 60, 100]

**Code:**



```
*cumulative sum.py - C:/chapter 5 lists/cumulative sum.py (3.6.5)*
File Edit Format Run Options Window Help
list=[1, 3, 5, 7, 8]
finallist=[list[0]]
for i in range(1, len(list)):
    finallist += [finallist[i-1]+list[i]]
print (finallist)
Ln: 1 Col: 0
```

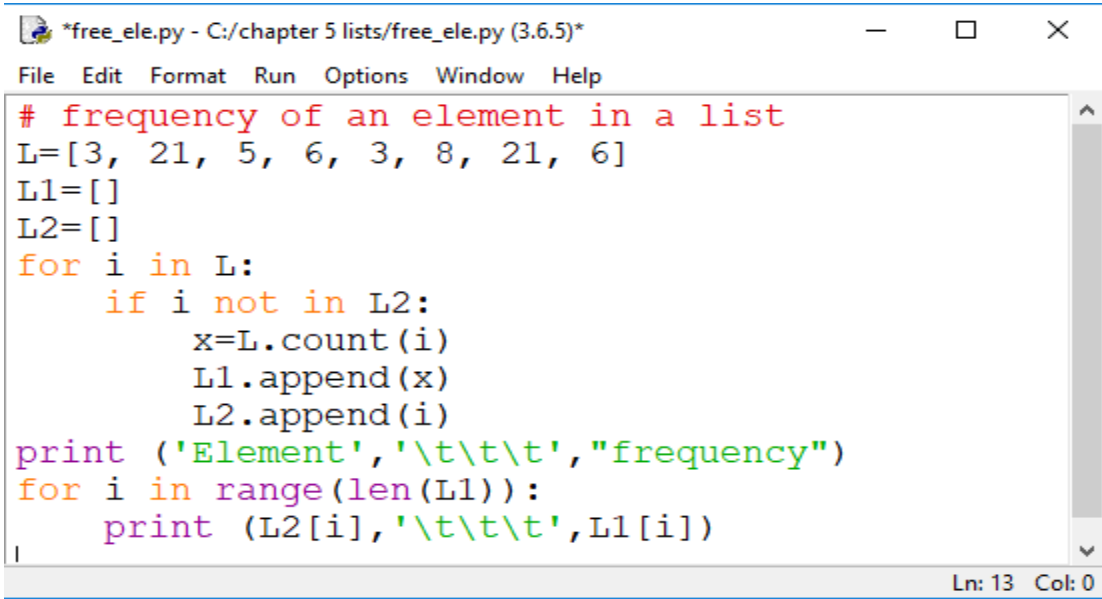
**\*\*\*\*\*Output of the program\*\*\*\*\***



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/chapter 5 lists/cumulative sum.py =====
[1, 4, 9, 16, 24]
>>> |
Ln: 6 Col: 4
```

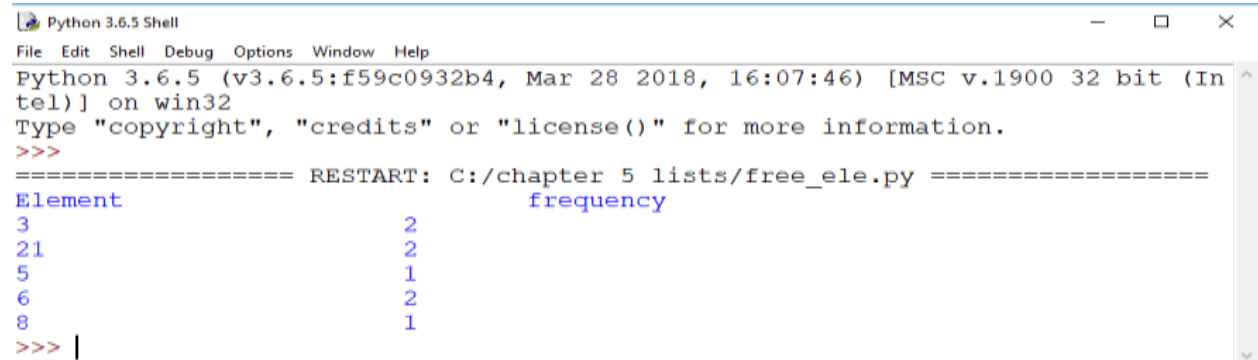
## Program 17 -WAP to display frequencies of all the elements of a list.

### Code:



```
*free_ele.py - C:/chapter 5 lists/free_ele.py (3.6.5)*
File Edit Format Run Options Window Help
# frequency of an element in a list
L=[3, 21, 5, 6, 3, 8, 21, 6]
L1=[]
L2=[]
for i in L:
    if i not in L2:
        x=L.count(i)
        L1.append(x)
        L2.append(i)
print ('Element','\t\t\t',"frequency")
for i in range(len(L1)):
    print (L2[i],'\t\t\t',L1[i])
Ln: 13 Col: 0
```

### \*\*\*\*\*Output of the program\*\*\*\*\*



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/chapter 5 lists/free_ele.py =====
Element                frequency
3                        2
21                       2
5                        1
6                        2
8                        1
>>> |
```

## Program 18 - WAP in Python to display those strings which are string with 'A' of given list.

### Code:

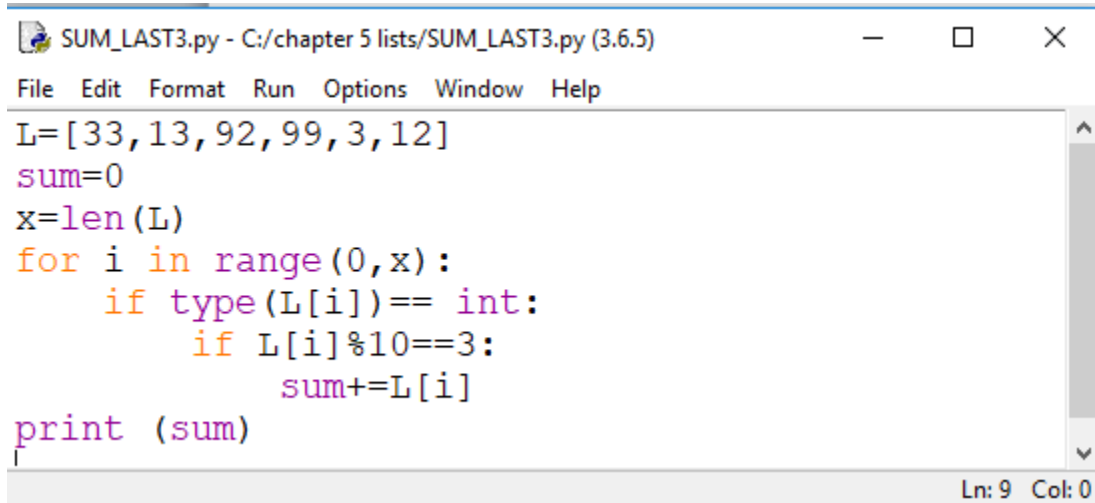
```
ELE_START_A.py - C:/chapter 5 lists/ELE_START_A.py (3.6.5)
File Edit Format Run Options Window Help
L=['AUSHIM', 'LEENA', 'AKHTAR', 'HIBA', 'NISHANT', 'AMAR']
count =0
for i in L:
    if i[0] in ('aA'):
        count+=1
        print (i)
print ("Appearing " ,count, " times")
Ln: 1 Col: 53
```

### \*\*\*\*\*Output of the program\*\*\*\*\*

```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/chapter 5 lists/ELE_START_A.py =====
AUSHIM
AKHTAR
AMAR
Appearing 3 times
Ln: 9 Col: 4
```

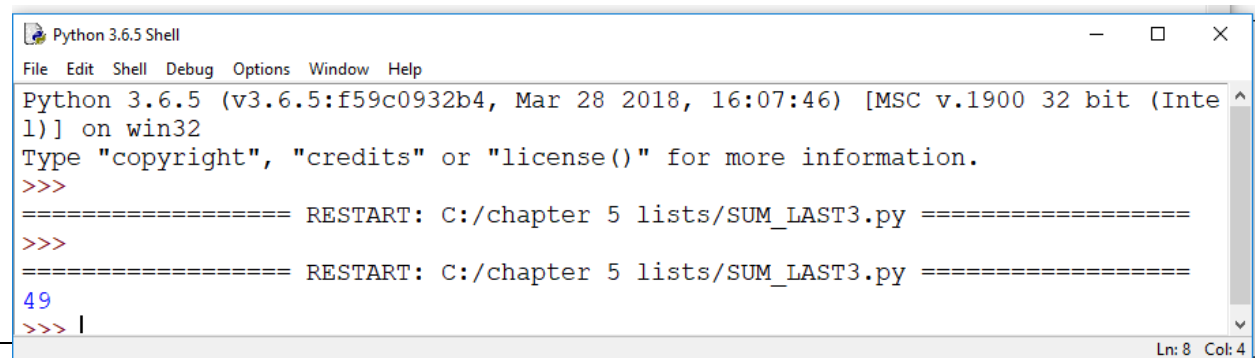
**Program 19 - WAP in Python to find and display the sum of all the values which are ending with 3 from a list.**

**Code:**



```
SUM_LAST3.py - C:/chapter 5 lists/SUM_LAST3.py (3.6.5)
File Edit Format Run Options Window Help
L=[33,13,92,99,3,12]
sum=0
x=len(L)
for i in range(0,x):
    if type(L[i])==int:
        if L[i]%10==3:
            sum+=L[i]
print (sum)
Ln: 9 Col: 0
```

**\*\*\*\*\*Output of the program\*\*\*\*\***



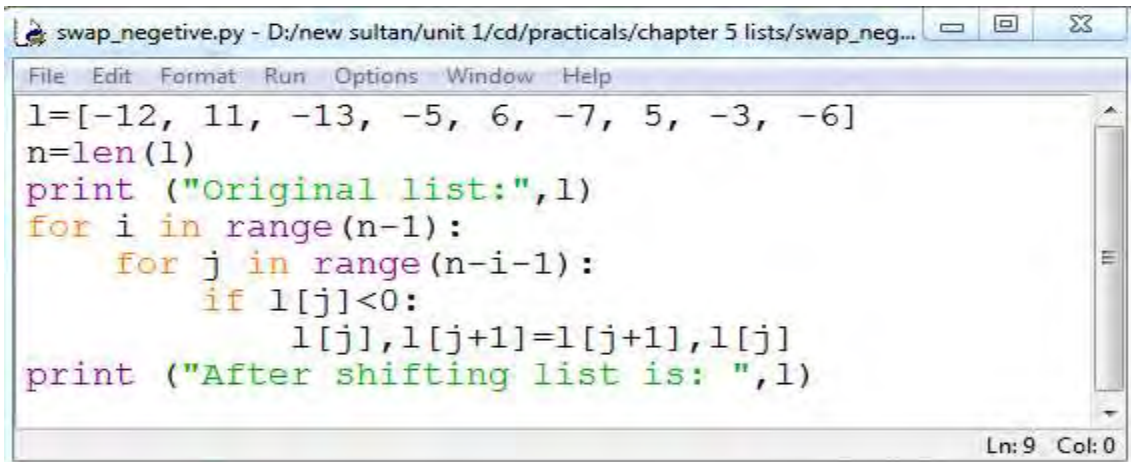
```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/chapter 5 lists/SUM_LAST3.py =====
>>>
===== RESTART: C:/chapter 5 lists/SUM_LAST3.py =====
49
>>> |
Ln: 8 Col: 4
```

**Program 20 - WAP to shift the negative number to left and the positive numbers to right so that the resultant list will look like.**

**Original list [-12, 11, -13, -5, 6, -7, 5, -3, -6]**

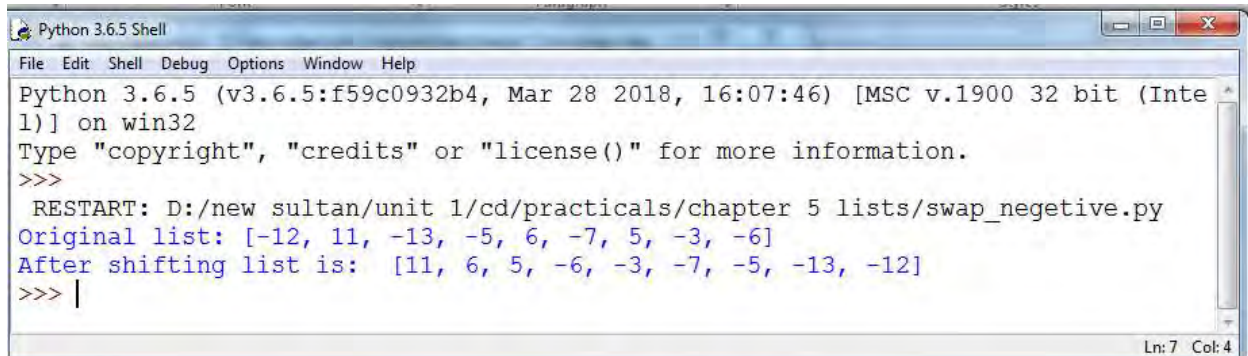
**Output should be [11, 6, 5, -6, -3, -7, -5, -13, -12]**

**Code:**



```
swap_negetive.py - D:/new sultan/unit 1/cd/practicals/chapter 5 lists/swap_neg...
File Edit Format Run Options Window Help
l=[-12, 11, -13, -5, 6, -7, 5, -3, -6]
n=len(l)
print ("Original list:",l)
for i in range(n-1):
    for j in range(n-i-1):
        if l[j]<0:
            l[j],l[j+1]=l[j+1],l[j]
print ("After shifting list is: ",l)
Ln: 9 Col: 0
```

**\*\*\*\*\*Output of the program\*\*\*\*\***



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:/new sultan/unit 1/cd/practicals/chapter 5 lists/swap_negetive.py
Original list: [-12, 11, -13, -5, 6, -7, 5, -3, -6]
After shifting list is: [11, 6, 5, -6, -3, -7, -5, -13, -12]
>>> |
Ln: 7 Col: 4
```

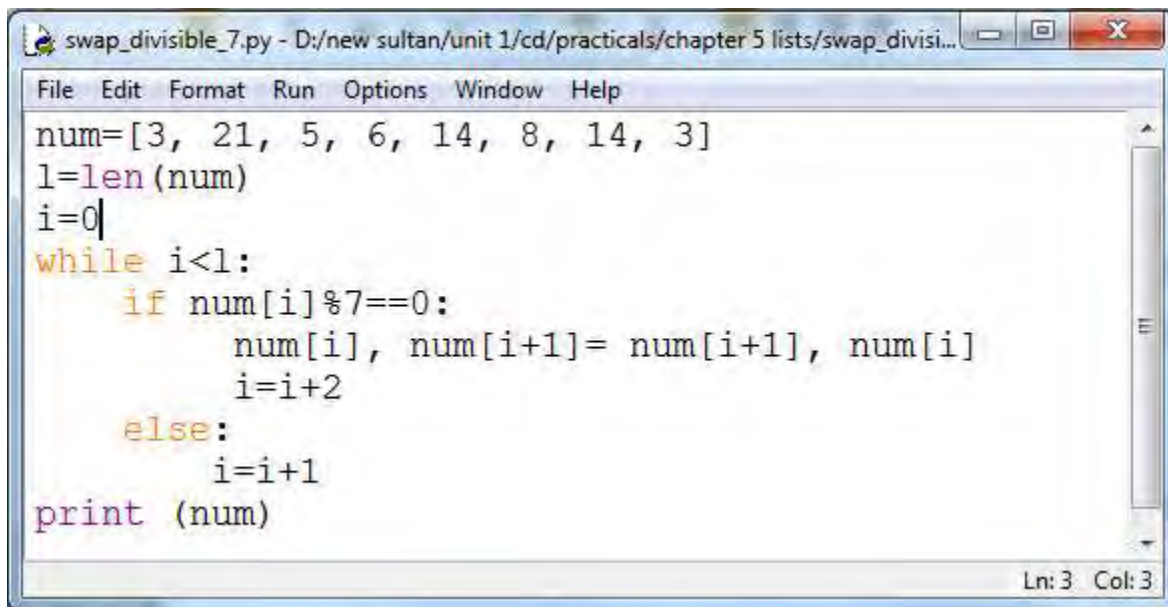
**Program 21-** A list Num contains the following elements:

3, 21, 5, 6, 14, 8, 14, 3

WAP to swap the content with next value divisible by 7 so that the resultant array will look like:

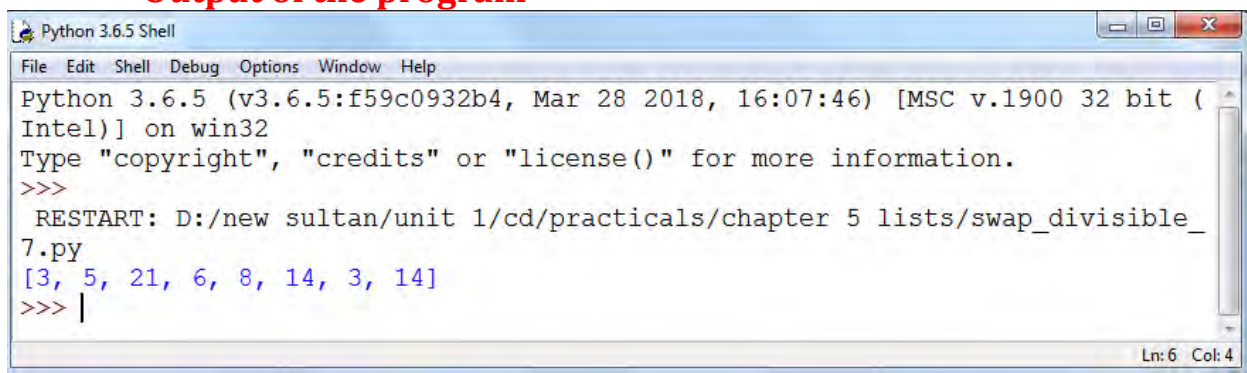
3, 5, 21, 6, 8, 14, 3, 14

**Code:**



```
swap_divisible_7.py - D:/new sultan/unit 1/cd/practicals/chapter 5 lists/swap_divisi...
File Edit Format Run Options Window Help
num=[3, 21, 5, 6, 14, 8, 14, 3]
l=len(num)
i=0
while i<l:
    if num[i]%7==0:
        num[i], num[i+1]= num[i+1], num[i]
        i=i+2
    else:
        i=i+1
print (num)
Ln: 3 Col: 3
```

**\*\*\*\*\*Output of the program\*\*\*\*\***

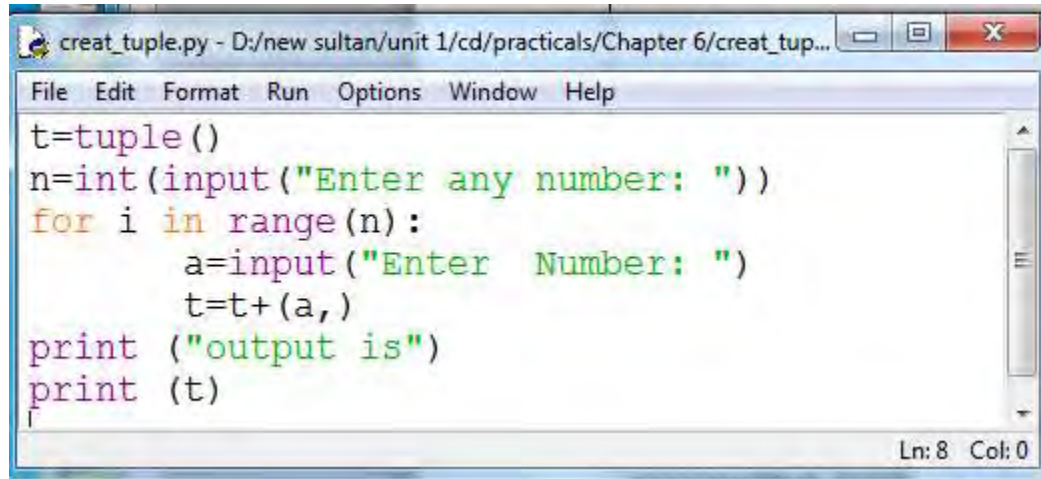


```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:/new sultan/unit 1/cd/practicals/chapter 5 lists/swap_divisible_7.py
[3, 5, 21, 6, 8, 14, 3, 14]
>>> |
Ln: 6 Col: 4
```



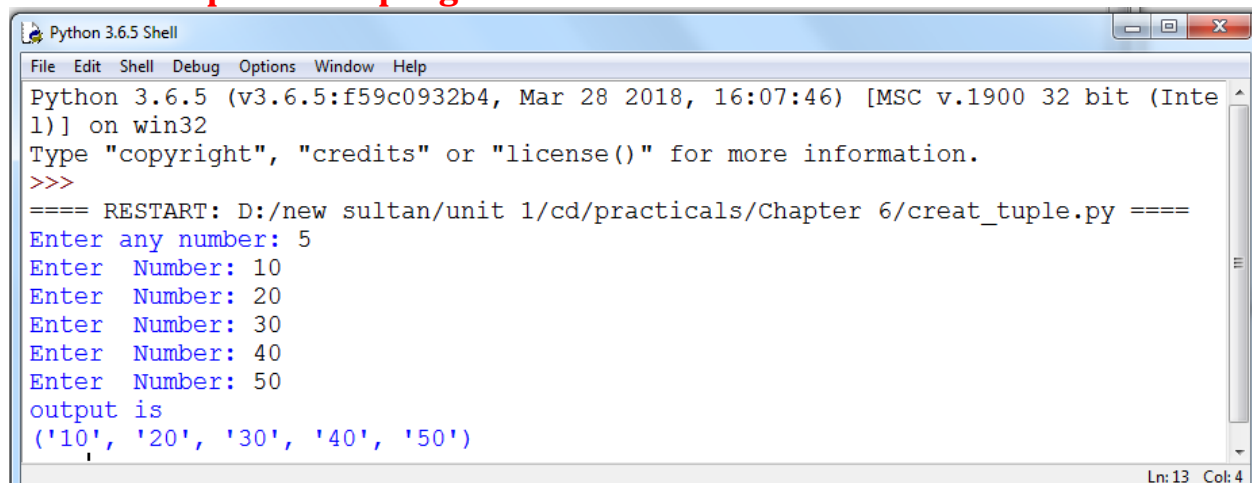
## Program 22-WAP to accept values from user and create a tuple.

### Code:



```
creat_tuple.py - D:/new sultan/unit 1/cd/practicals/Chapter 6/creat_tup...
File Edit Format Run Options Window Help
t=tuple()
n=int(input("Enter any number: "))
for i in range(n):
    a=input("Enter Number: ")
    t=t+(a,)
print ("output is")
print (t)
Ln: 8 Col: 0
```

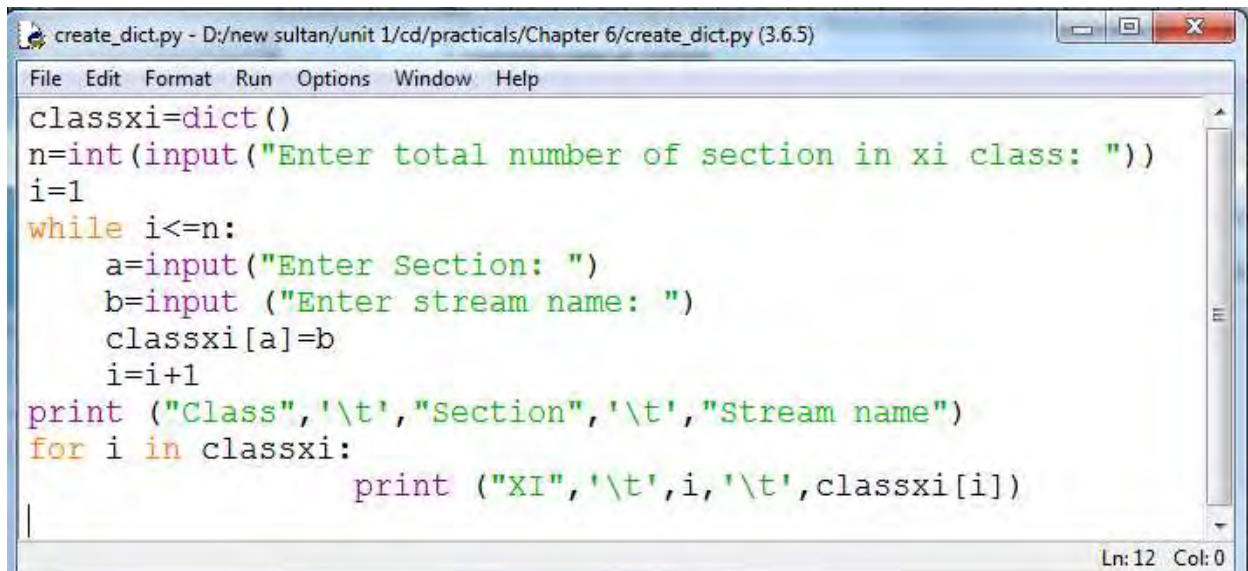
### \*\*\*\*\*Output of the program\*\*\*\*\*



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: D:/new sultan/unit 1/cd/practicals/Chapter 6/creat_tuple.py ====
Enter any number: 5
Enter Number: 10
Enter Number: 20
Enter Number: 30
Enter Number: 40
Enter Number: 50
output is
('10', '20', '30', '40', '50')
Ln: 13 Col: 4
```

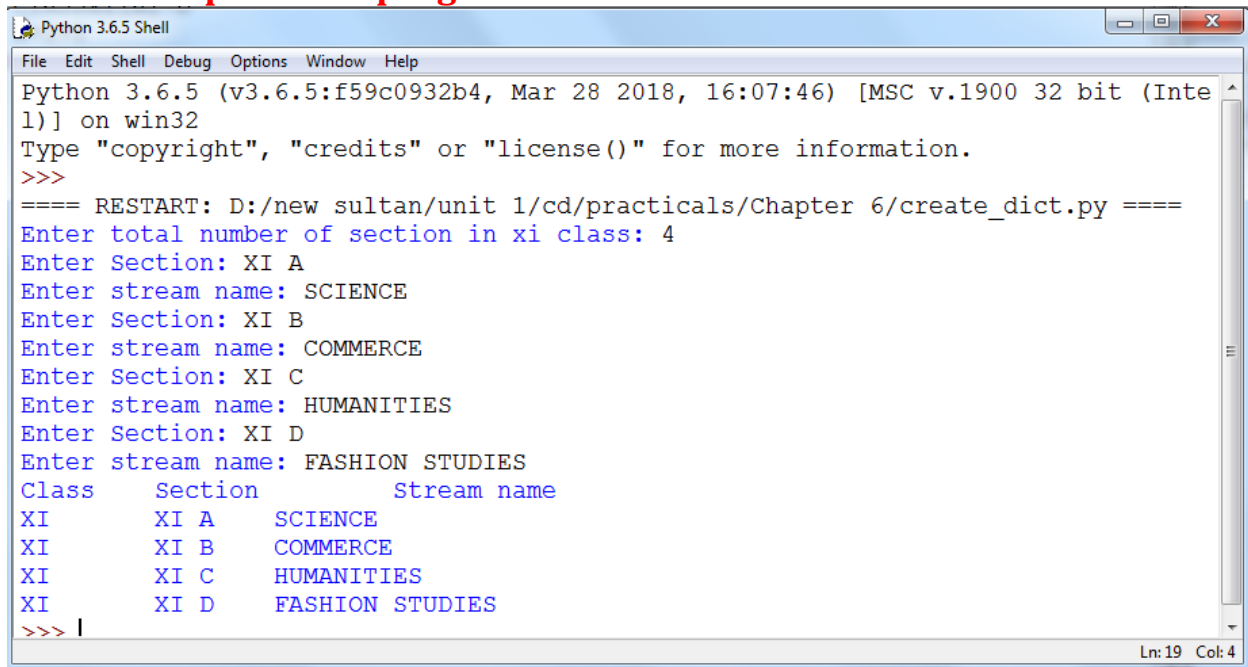
**Program 23-** Write a program to input total number of sections and stream name in 11<sup>th</sup> class and display all information on the output screen.

**Code:**



```
create_dict.py - D:/new sultan/unit 1/cd/practicals/Chapter 6/create_dict.py (3.6.5)
File Edit Format Run Options Window Help
classxi=dict()
n=int(input("Enter total number of section in xi class: "))
i=1
while i<=n:
    a=input("Enter Section: ")
    b=input("Enter stream name: ")
    classxi[a]=b
    i=i+1
print("Class","\t","Section","\t","Stream name")
for i in classxi:
    print("XI","\t",i,"\t",classxi[i])
Ln:12 Col:0
```

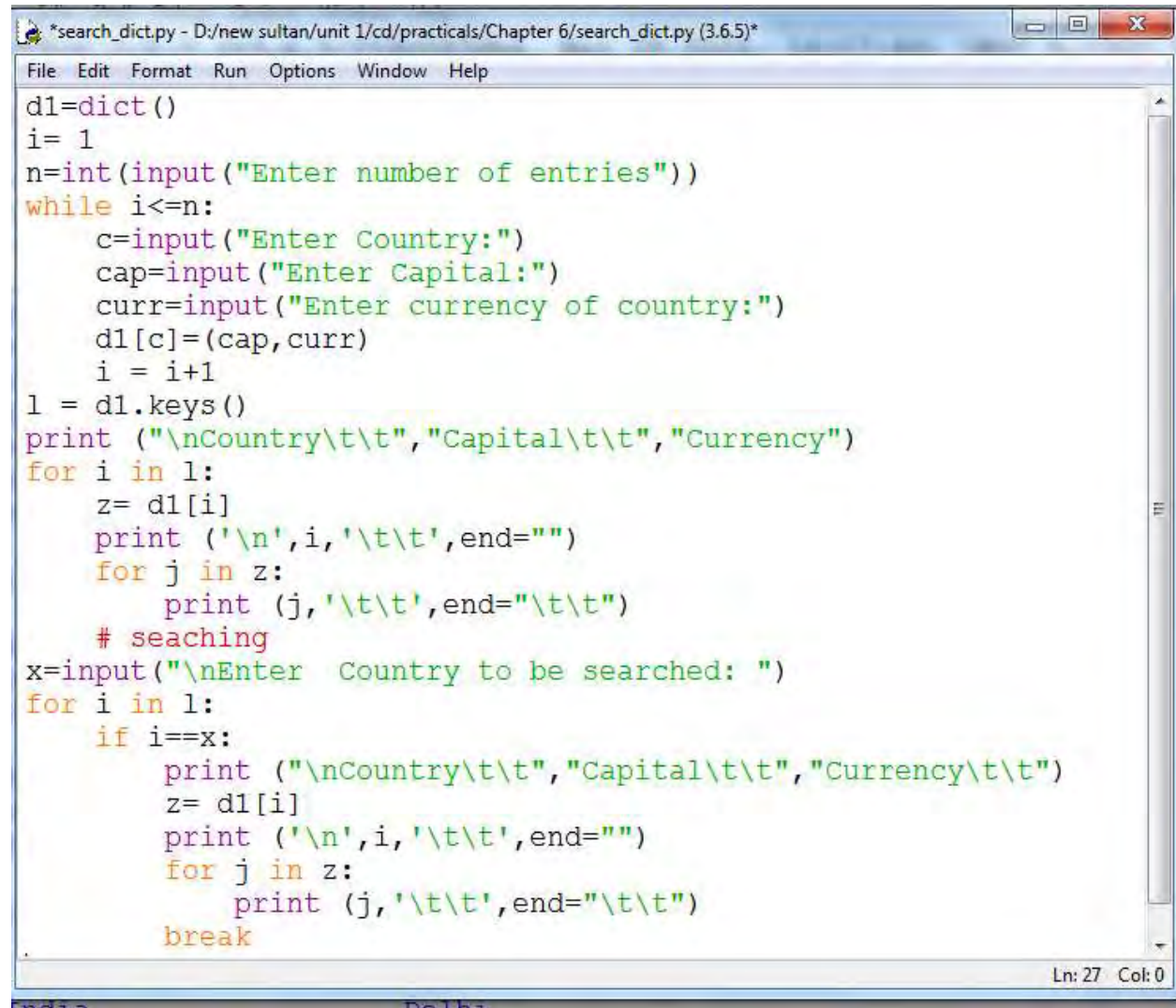
**\*\*\*\*\*Output of the program\*\*\*\*\***



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: D:/new sultan/unit 1/cd/practicals/Chapter 6/create_dict.py ====
Enter total number of section in xi class: 4
Enter Section: XI A
Enter stream name: SCIENCE
Enter Section: XI B
Enter stream name: COMMERCE
Enter Section: XI C
Enter stream name: HUMANITIES
Enter Section: XI D
Enter stream name: FASHION STUDIES
Class      Section      Stream name
XI         XI A         SCIENCE
XI         XI B         COMMERCE
XI         XI C         HUMANITIES
XI         XI D         FASHION STUDIES
>>> |
Ln:19 Col:4
```

**Program 24-** Write a Python program to input names of 'n' countries and their capital and currency, store it in a dictionary and display in tabular form. Also search and display for a particular country.

**Code:**



```
*search_dict.py - D:/new sultan/unit 1/cd/practicals/Chapter 6/search_dict.py (3.6.5)*
File Edit Format Run Options Window Help
d1=dict()
i= 1
n=int(input("Enter number of entries"))
while i<=n:
    c=input("Enter Country:")
    cap=input("Enter Capital:")
    curr=input("Enter currency of country:")
    d1[c]=(cap,curr)
    i = i+1
l = d1.keys()
print ("\nCountry\t\t", "Capital\t\t", "Currency")
for i in l:
    z= d1[i]
    print ('\n',i, '\t\t',end="")
    for j in z:
        print (j, '\t\t',end="\t\t")
    # seaching
x=input("\nEnter Country to be searched: ")
for i in l:
    if i==x:
        print ("\nCountry\t\t", "Capital\t\t", "Currency\t\t")
        z= d1[i]
        print ('\n',i, '\t\t',end="")
        for j in z:
            print (j, '\t\t',end="\t\t")
        break
Ln: 27 Col: 0
```

**\*\*\*\*\*Output of the program\*\*\*\*\***

```
Enter number of entries3
Enter Country:Austria
Enter Capital:Vienna
Enter currency of country:Euro
Enter Country:India
Enter Capital:Delhi
Enter currency of country:Rupee
Enter Country:France
Enter Capital:Paris
Enter currency of country:Euro
```

Country	Capital	Currency
Austria	Vienna	Euro
India	New Delhi	Indian Rupee
France	Paris	Euro
Enter Country to be searched:	India	
Country	Capital	Currency
India	New Delhi	Indian Rupee



# MY SQL

(30 queries)

1. Command for creating a database.

```
mysql> create database students;
Query OK, 1 row affected (0.00 sec)
```

2. Command for using the database.

```
mysql> use students;
Database changed
mysql>
```

3. Command for creating a table.

```
mysql> create table student(admno int,
-> name varchar(30),
-> class int,
-> sec char,
-> rno int,
-> address varchar(30));
Query OK, 0 rows affected (0.05 sec)
```

4. Command for showing the structure of table.

```
mysql> desc student;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| admno | int(11) | YES | | NULL | |
| name | varchar(30) | YES | | NULL | |
| class | int(11) | YES | | NULL | |
| sec | char(1) | YES | | NULL | |
| rno | int(11) | YES | | NULL | |
| address | varchar(30) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

5. Command to show tables present in database.

```
mysql> show tables;
+-----+
| Tables_in_students |
+-----+
| student |
+-----+
1 row in set (0.00 sec)
```

## 6. Command for inserting data into a table.

```
mysql> insert into student values (1234,"Aditi sharma",9,"A",4,"SJE");
Query OK, 1 row affected (0.06 sec)

mysql> insert into student values (2605,"shreya nagpal",10,"D",7,"Jor bagh");
Query OK, 1 row affected (0.06 sec)

mysql> insert into student values (3712,"tanya verma",11,"C",21,"Malviya nagar");
Query OK, 1 row affected (0.05 sec)

mysql> insert into student values (5612,"krish gupta",12,"B",15,"Janak puri");
Query OK, 1 row affected (0.05 sec)

mysql> insert into student values (6523,"zayn malik",11,"E",40,"Rohini");
Query OK, 1 row affected (0.06 sec)

mysql> insert into student values (4031,"shivani mehta",9,"A",33,"Hauz khas");
Query OK, 1 row affected (0.05 sec)
```

## 7. Command to view the contents of the table.

```
mysql> select * from student;
```

admno	name	class	sec	eno	address
1234	Aditi sharma	9	A	4	SJE
2605	shreya nagpal	10	D	7	Jor bagh
3712	tanya verma	11	C	21	Malviya nagar
5612	krish gupta	12	B	15	Janak puri
6523	zayn malik	11	E	40	Rohini
4031	shivani mehta	9	A	33	Hauz khas

```
6 rows in set (0.00 sec)
```

## 8. Command to retrieve data.

```
mysql> select name , class from student;
```

name	class
Aditi sharma	9
shreya nagpal	10
tanya verma	11
krish gupta	12
zayn malik	11
shivani mehta	9

```
6 rows in set (0.00 sec)
```

9. Command for using keyword DISTINCT.

```
mysql> select DISTINCT sec from student;
+----+
| sec |
+----+
| A   |
| D   |
| C   |
| B   |
| E   |
+----+
5 rows in set (0.00 sec)
```

10. Command for using WHERE clause.

```
mysql> select name, class from student where class>9;
+-----+-----+
| name      | class |
+-----+-----+
| shreya nagpal | 10    |
| tanya verma  | 11    |
| krish gupta  | 12    |
| zayn malik   | 11    |
+-----+-----+
4 rows in set (0.00 sec)
```

11. Command for using ORDER BY clause.

```
mysql> select * from student order by class,name;
+-----+-----+-----+-----+-----+-----+
| admno | name           | class | sec | rno | address      |
+-----+-----+-----+-----+-----+-----+
| 1234  | Aditi sharma  | 9     | A   | 4   | SJE          |
| 4031  | shivani mehta | 9     | A   | 33  | Hauz khas   |
| 2605  | shreya nagpal | 10    | D   | 7   | Jor bagh    |
| 3712  | tanya verma   | 11    | C   | 21  | Malviya nagar |
| 6523  | zayn malik    | 11    | E   | 40  | Rohini      |
| 5612  | krish gupta   | 12    | B   | 15  | Janak puri  |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

12. Command for using UPDATE .

```
mysql> update student set class=10 where name='shreya nagpal';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0

mysql> alter table student add marks int;
Query OK, 6 rows affected (0.06 sec)
Records: 6 Duplicates: 0 Warnings: 0
```



13. Command for using ALTER (to modify structure of table).

```
mysql> alter table student add marks int;
Query OK, 6 rows affected (0.06 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

14. Command for using LIKE operator.

```
mysql> select name,class from student where name like 's%';
+-----+-----+
| name      | class |
+-----+-----+
| shreya nagpal | 10    |
| shivani mehta | 9     |
+-----+-----+
2 rows in set (0.01 sec)
```

15. Command for using aggregate functions.

```
mysql> select max(class),min(admno) from student;
+-----+-----+
| max(class) | min(admno) |
+-----+-----+
| 12         | 1234       |
+-----+-----+
1 row in set (0.06 sec)
```

16. Command for using GROUP BY.

```
mysql> select min(class),name from student group by name;
+-----+-----+
| min(class) | name      |
+-----+-----+
| 9          | Aditi sharma |
| 12         | krish gupta  |
| 9          | shivani mehta |
| 10         | shreya nagpal |
| 11         | tanya verma  |
| 11         | zayn malik   |
+-----+-----+
6 rows in set (0.00 sec)
```

17. Command for using HAVING clause.

```
mysql> select avg(admno),class from student group by class having avg(admno)>2000;
```

avg(admno)	class
2632.5000	9
2605.0000	10
5117.5000	11
5612.0000	12

4 rows in set (0.08 sec)

18. Command for using Group by with order by.

```
mysql> select sum(class),sec from student group by class order by class desc;
```

sum(class)	sec
12	B
22	C
10	D
18	A

4 rows in set (0.06 sec)

19. Command for using group by and having clause with where clause.

```
mysql> select avg(marks),name,class from student where class in(10,11) group by name having avg(marks)<2000;
```

avg(marks)	name	class
25.0000	shreya nagpal	10
17.0000	tanya verma	11
40.0000	zayn malik	11

3 rows in set (0.00 sec)

20. Command for equi-join of tables.

```
mysql> select * from student,sports where student.admno=sports.admno;
```

admno	name	class	sec	rno	address	marks	admno
6523	zayn malik	11	E	40	Rohini	40	6523
4031	shivani mehta	9	A	33	Hauz khas	55	4031
3712	tanya verma	11	C	21	Malviya nagar	17	3712
5612	krish gupta	12	B	15	Janak puri	35	5612
1234	Aditi sharma	9	A	4	SJE	30	1234
2605	shreya nagpal	10	D	7	Jor bagh	25	2605

6 rows in set (0.00 sec)

21. Command to retrieve data from two tables.

```
mysql> select name,grade from student,sports where student.admno=sports.admno and coach_name="SK Singh";
```

name	grade
shivani mehta	A
tanya verma	B

2 rows in set (0.00 sec)

22. Command for using group by clause in join.

```
mysql> select avg(marks),grade from student,sports where student.admno=sports.admno group by grade;
```

avg(marks)	grade
40.0000	A
23.5000	B
35.0000	C

3 rows in set (0.00 sec)

```
mysql> select sum(marks),grade from student,sports where student.admno=sports.admno group by address order by sum(marks) desc;
```

sum(marks)	grade
55	A
40	A
35	C
30	B
25	A
17	B

6 rows in set (0.00 sec)

23. Command for using group by and order by clause in equi-join.

24. Command for using where clause and group by.

```
mysql> select name,class from student where marks>10 group by class;
```

name	class
Aditi sharma	9
shreya nagpal	10
tanya verma	11
krish gupta	12

4 rows in set (0.00 sec)

25. Command for adding primary key.

```
mysql> alter table student add primary key(admno);
Query OK, 6 rows affected (0.06 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

26. Command to delete a column.

```
mysql> alter table student drop rno;
Query OK, 6 rows affected (0.06 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

27. Command to remove primary key.

```
mysql> alter table student drop primary key;
Query OK, 6 rows affected (0.13 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

28. Command to increase marks.

```
mysql> update student set marks=marks+10;
Query OK, 6 rows affected (0.05 sec)
Rows matched: 6 Changed: 6 Warnings: 0
```

29. Command to change data type of an existing column.

```
mysql> alter table student modify marks decimal(8,2);
Query OK, 6 rows affected (0.06 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

30. Command to a delete table.

```
mysql> drop table sports;
Query OK, 0 rows affected (0.05 sec)
```