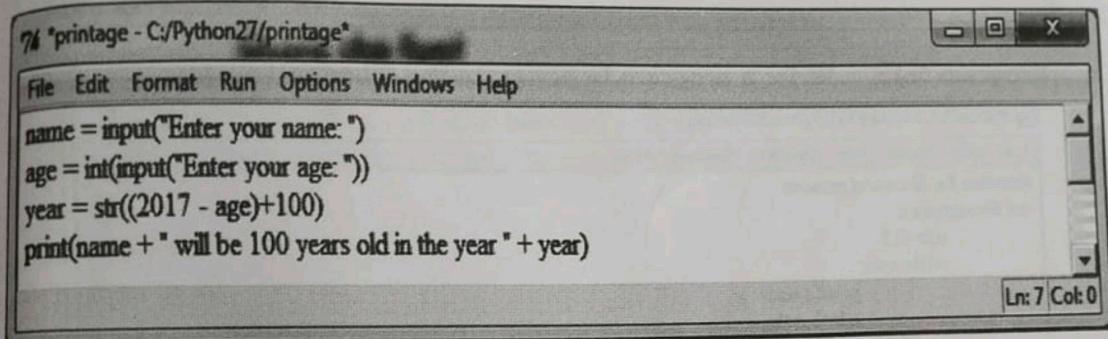


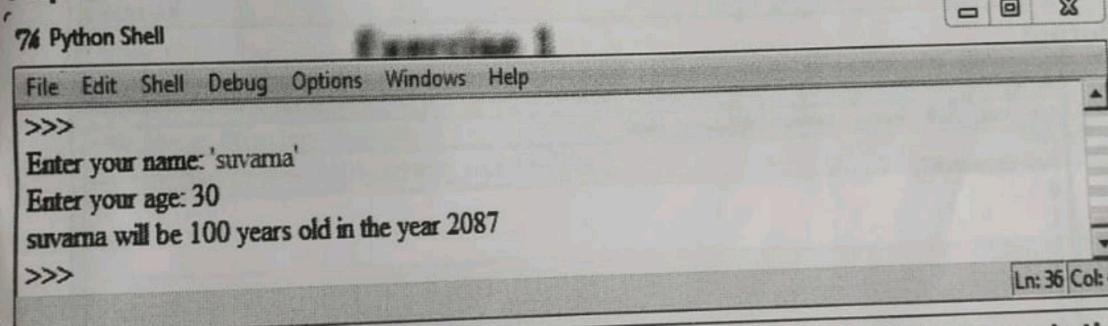
List of Practicals

- **Program 1(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.



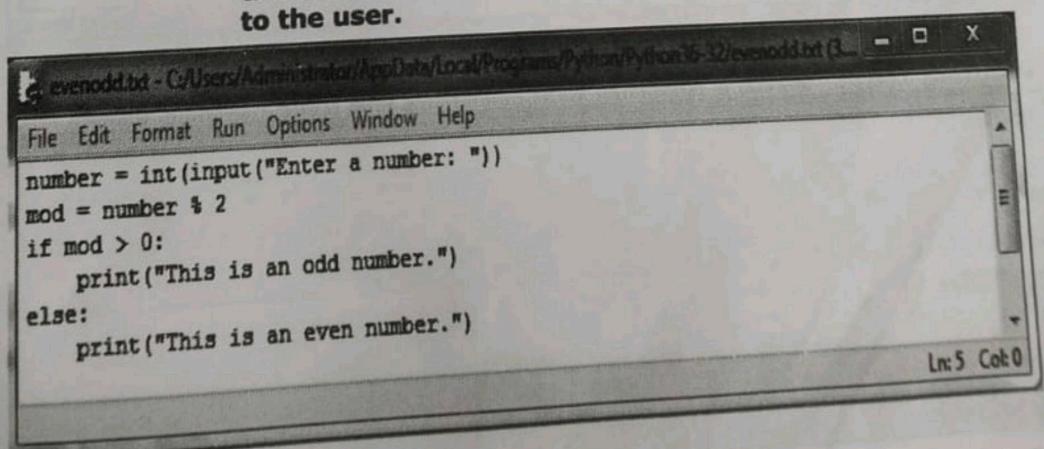
```
% *printage - C:/Python27/printage*
File Edit Format Run Options Windows Help
name = input("Enter your name: ")
age = int(input("Enter your age: "))
year = str((2017 - age)+100)
print(name + " will be 100 years old in the year " + year)
Ln: 7 Col: 0
```

Output

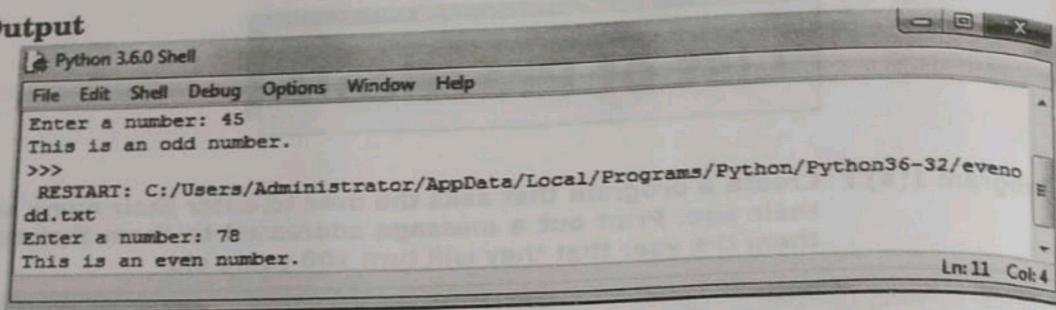


```
% Python Shell
File Edit Shell Debug Options Windows Help
>>>
Enter your name: 'suvama'
Enter your age: 30
suvama will be 100 years old in the year 2087
>>>
Ln: 36 Col: 4
```

- **Program 1(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.

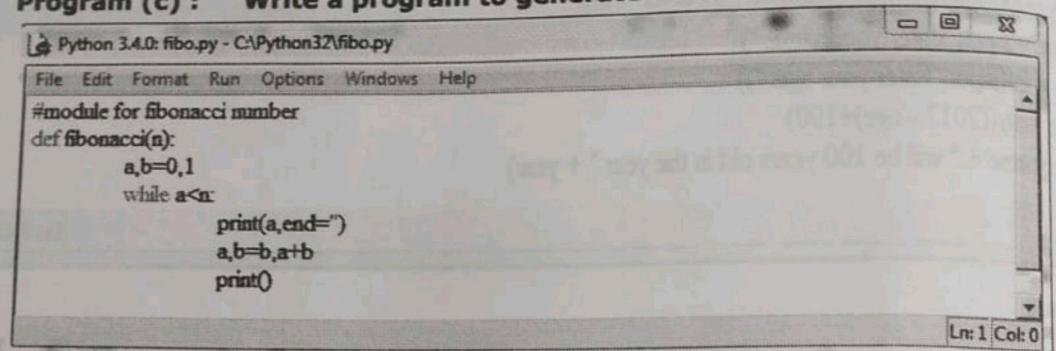


```
evenodd.txt - C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32/evenodd.txt
File Edit Format Run Options Window Help
number = int(input("Enter a number: "))
mod = number % 2
if mod > 0:
    print("This is an odd number.")
else:
    print("This is an even number.")
Ln: 5 Col: 0
```

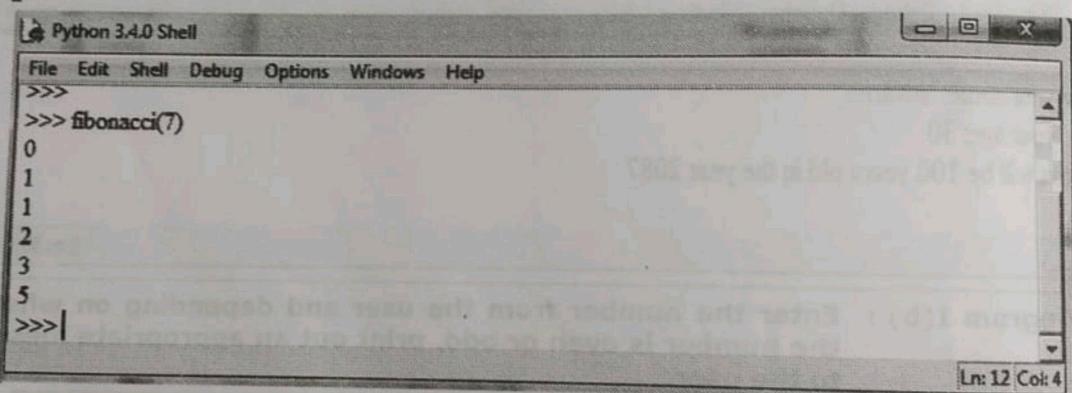
Output


```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Enter a number: 45
This is an odd number.
>>>
RESTART: C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32/evenodd.txt
Enter a number: 78
This is an even number.
Ln: 11 Col: 4
```

- > **Program (c) :** Write a program to generate the Fibonacci series.



```
Python 3.4.0: fibo.py - C:\Python32\fibo.py
File Edit Format Run Options Windows Help
#module for fibonacci number
def fibonacci(n):
    a,b=0,1
    while a<n:
        print(a,end=" ")
        a,b=b,a+b
        print()
```

Output


```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
>>>
>>> fibonacci(7)
0
1
1
2
3
5
>>> | Ln: 12 Col: 4
```

- > **Program 1(d) :** Write a function that reverses the user defined value.

```
def rev(s):
    r=""
    i=len(s)
    while i>0:
        r+=str1[index-1]
        i=i+1
    return r
print(rev('python'))
```

Output

nohtyp

Program 1(e) : Write a function to check the input value is Armstrong and also write the function for Palindrome.

Armstrong

(1)

```
def As():
    n = input('Enter Number to check for Armstrong')
    d = input('Your Entered number is of how many digits')
    f = n
    s = 0
    while(f!=0):
        a = f % 10
        f = f / 10
        s = s + ( a ** d)

    if( s == n):
        print('%d is a armstrong number' %n)
    else:
        print('%d is not a armstrong number' %n)
```

Output

```
>>> As()
Enter Number to check for Armstrong153
Your Entered number is of how many digits3
153 is a armstrong number
```

```
>>> As()
Enter Number to check for Armstrong121
Your Entered number is of how many digits3
121 is not a armstrong number
```

>>>

(2)

```
# Python program to determine whether the number is
# Armstrong number or not
```



```
# Function to calculate x raised to the power y
Def power (x,y);
    If y==0;
        Return 1
    If y%2==0;
        Return power(x, y/2)*power(x, y/2)
        Return x*power(x, y/2)*power(x, y/2)
# Function to calculate order of the number
Def order (x);

# Variable to store of the number
n=0
while (x!=0);
    n=n+1
    x=x/10
return n

# Function to check whether the given number is
# Armstrong number or not
Def is Armstrong (x);
    N=order(x);
    Temp=x;
    Sum1=0;
    While(temp!=0);
        r=temp%10
        sum1=sum1+power (r,n)
        temp=temp/10

# If condition satisfies
    Return (sum1==x)

#Driver program
X=153
Print (isArmstrong (x))
X=1253
Print(isArmstrong (x))
```

Output

```
1
0
```

Program 1(f) : Write a recursive function to print the factorial for a given number.

```
def rec(n):
    if n==1:
        return n
    else:
        return n*rec(n-1)
num=7
if num<0:
    print("factorial does not exist for negative numbers")
elif num==0:
    print("the factorial of 0 is 1")
else:
    print("the factorial of",num,"is",rec(num))
```

Output

The factorial of 7 is 5040

Program 2(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.

```
def checkc():
    if(c=="a" or c=="A" or
    c=="e" or c=="E" or
    c=="i" or c=="I" or
    c=="o" or c=="O" or
    c=="u" or c=="U"):
        return "True"
    else:
        return "False"
Print("Enter the string to check")
c=input()
if check(c)=="True":
    Print("The character is Vowel")
else:
    Print("The character is not Vowel")
```

Output

```
def vowelChecker(a)
The entered character is Vowel.
```

- > **Program 2(b) :** Define a function that computes the length of a given list or string.

```
def countstr(inp):
    #DocString

    c = 0

    for i in inp:
        c = c+1

    print ("Length of the String is " +str(c))

    print ("Function to find the length of the input String")
    print ("Enter the string")

    inp = input()

    countstr(inp)
```

Output

```
>>>countstr('python')
Length of the String is 5
```

- > **Program 2(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 :

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

Python 3.4.0: histogram.py - C:/Python32/histogram.py*

File Edit Format Run Options Windows Help

```
#program for histogram
def histogram(inputList):

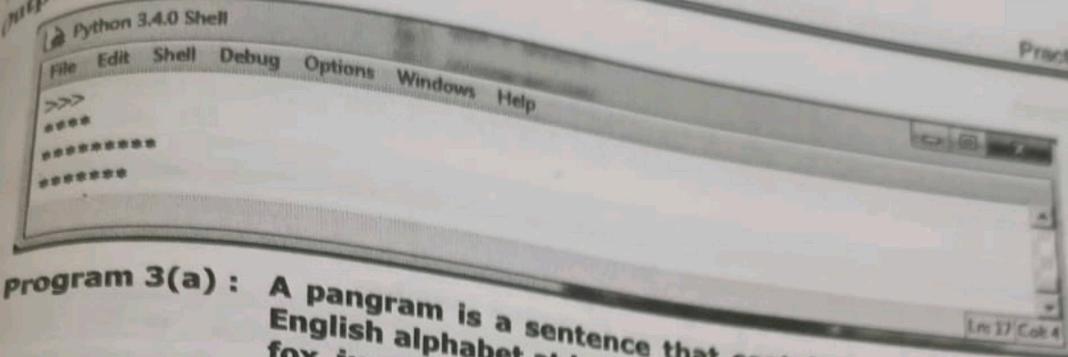
    for i in range(len(inputList)):
        print (inputList[i]***)
```

List = [4,9,7]

histogram(List)

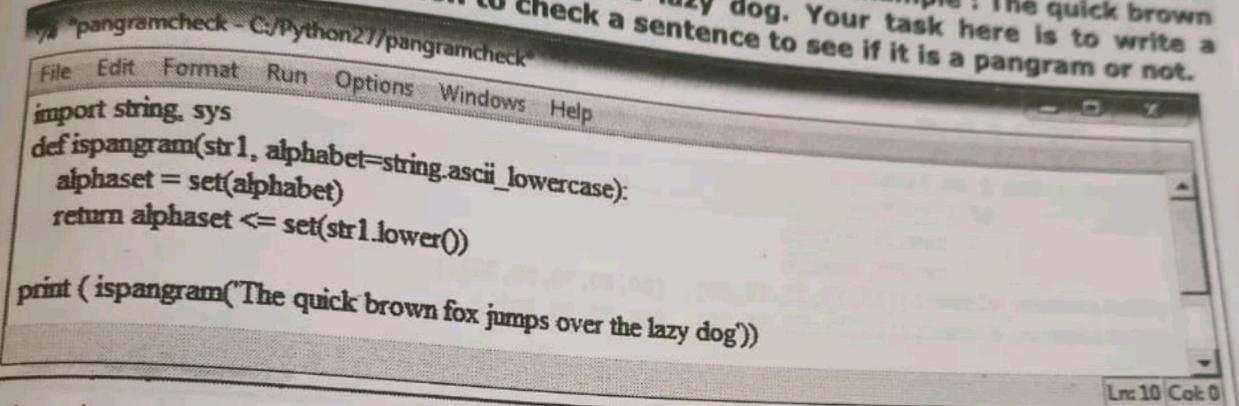
Practices
a given list or

Output



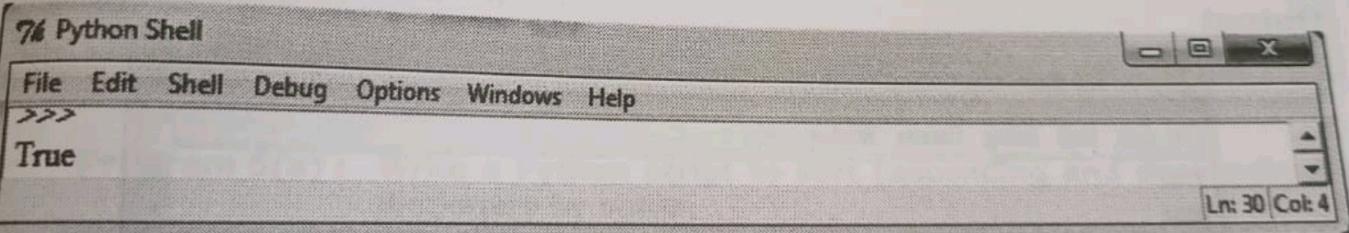
```
>>>
*****
*****
```

Program 3(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.



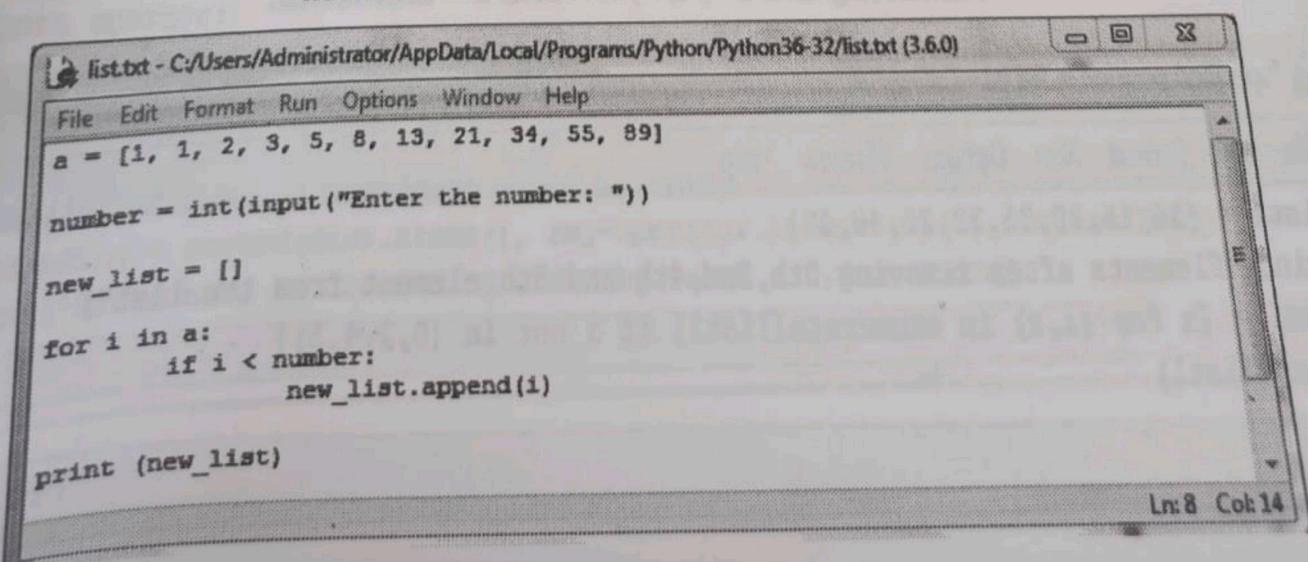
```
/*pangramcheck - C:/Python27/pangramcheck*/
File Edit Format Run Options Windows Help
import string, sys
def ispanagram(str1, alphabet=string.ascii_lowercase):
    alphaset = set(alphabet)
    return alphaset <= set(str1.lower())
print ( ispanagram('The quick brown fox jumps over the lazy dog'))
```

Output

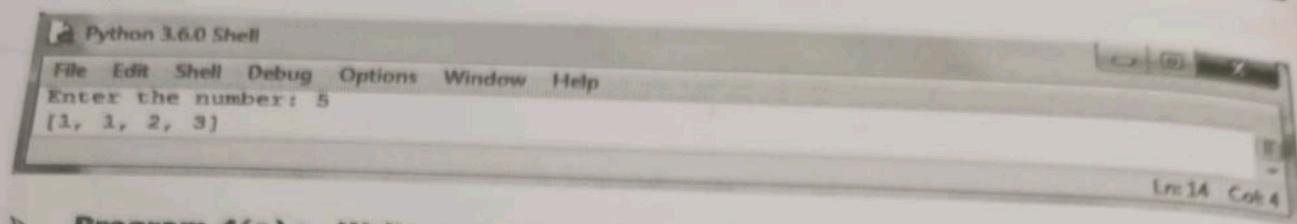


```
% Python Shell
File Edit Shell Debug Options Windows Help
>>>
True
```

➤ **Program 3(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

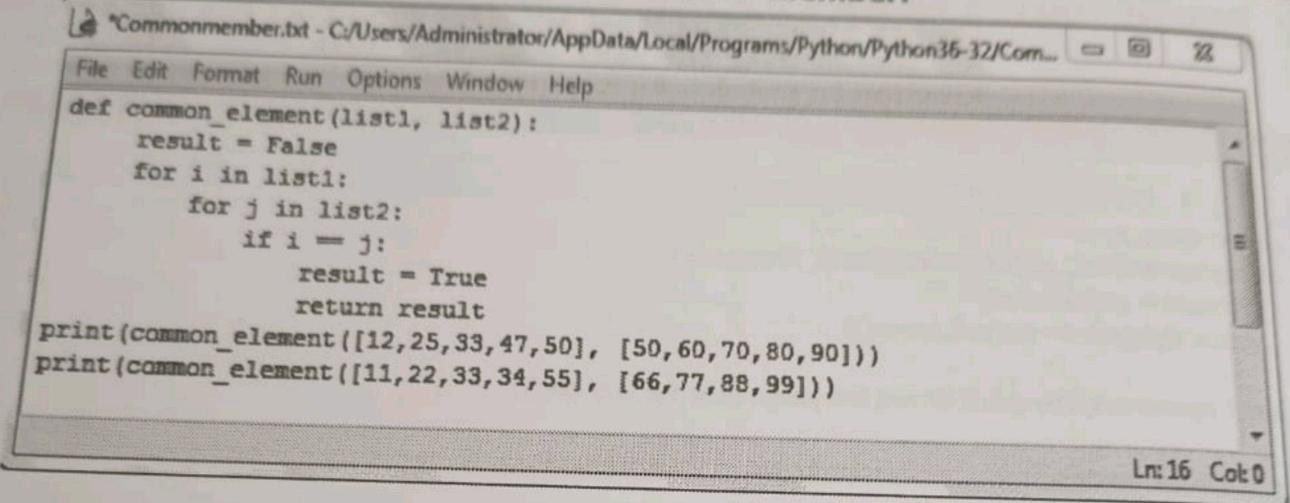


```
list.txt - C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32/list.txt (3.6.0)
File Edit Format Run Options Window Help
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
number = int(input("Enter the number: "))
new_list = []
for i in a:
    if i < number:
        new_list.append(i)
print (new_list)
```

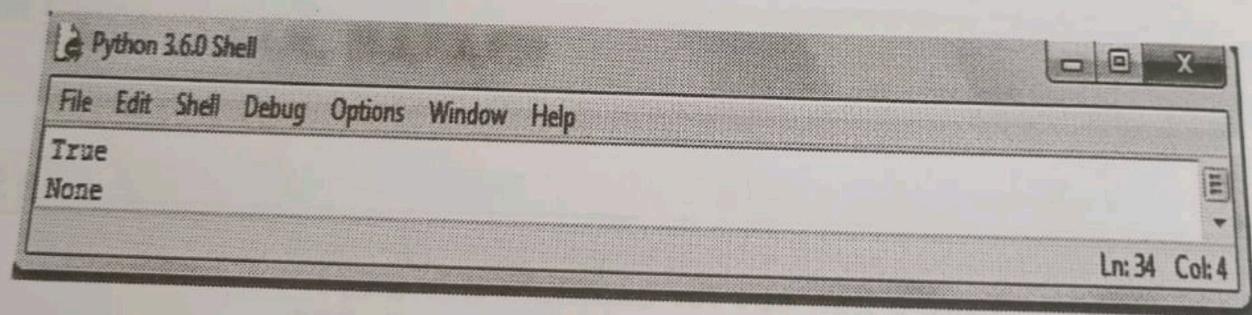
Output


```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Enter the number: 5
[1, 1, 2, 3]
Ln: 14 Col: 4
```

- **Program 4(a) :** Write a program that takes two lists and returns True if they have at least one Common member.

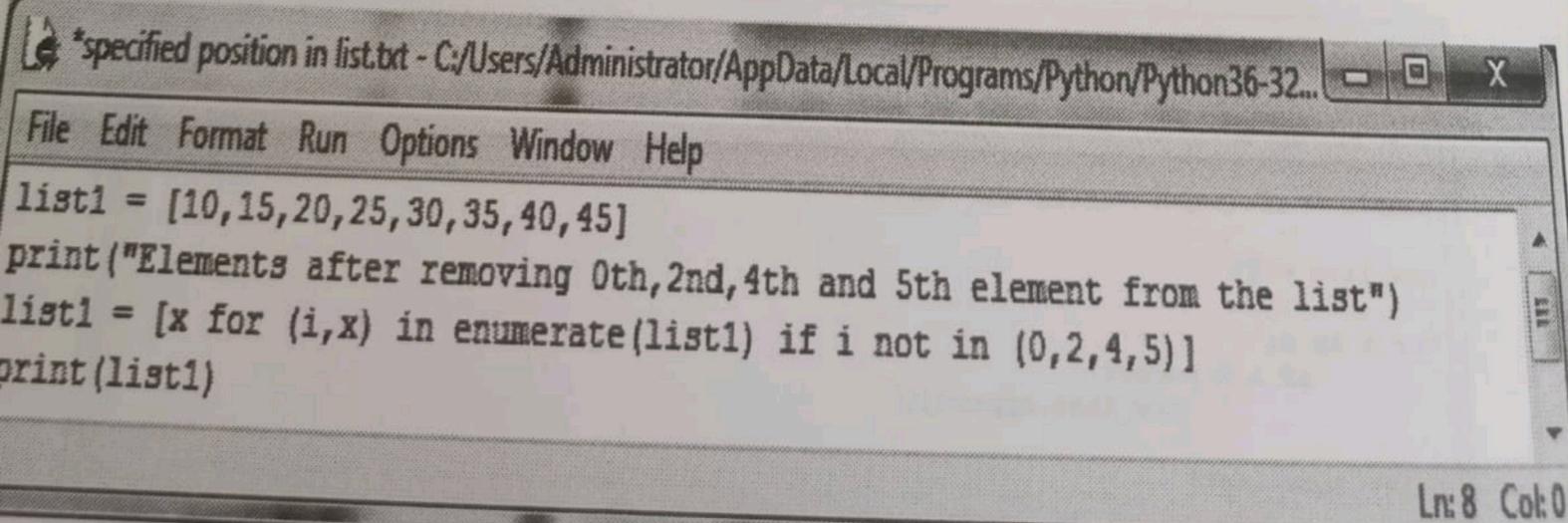


```
File Edit Format Run Options Window Help
def common_element(list1, list2):
    result = False
    for i in list1:
        for j in list2:
            if i == j:
                result = True
    return result
print(common_element([12,25,33,47,50], [50,60,70,80,90]))
print(common_element([11,22,33,34,55], [66,77,88,99]))
Ln: 16 Col: 0
```

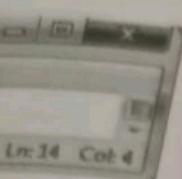
Output


```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
True
None
Ln: 34 Col: 4
```

- **Program 4(b) :** Write a Python program to print a specified list after removing the 0th, 2nd, 4th and 5th elements.



```
*specified position in list.txt - C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32...
File Edit Format Run Options Window Help
list1 = [10,15,20,25,30,35,40,45]
print("Elements after removing 0th,2nd,4th and 5th element from the list")
list1 = [x for (i,x) in enumerate(list1) if i not in (0,2,4,5)]
print(list1)
Ln: 8 Col: 0
```



ue if they

x

Output

Python 3.6.0 Shell

File Edit Shell Debug Options Window Help

Elements after removing 0th, 2nd, 4th and 5th element from the list
 [15, 25, 40, 45]

>>>

program 4(c) : Write a Python program to clone or copy a list

listclone.txt - C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32/

File Edit Format Run Options Window Help

```
list1 = [10, 20, 30, 40, 50, 60, 70, 80]
clonelist = list(list1)
print(list1)
print(clonelist)
```

Ln: 41 Col: 4

Output

Python 3.6.0 Shell

File Edit Shell Debug Options Window Help

[10, 20, 30, 40, 50, 60, 70, 80]
 [10, 20, 30, 40, 50, 60, 70, 80]

Ln: 18 Col: 4

- **Program 5(a) : Write a Python script to sort (ascending and descending) a dictionary by value.**

*sortingdictionary.txt - C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32/sortin...

File Edit Format Run Options Window Help

```
import operator
dict = {1: 12, 3: 14, 4: 13, 2: 11, 0: 10}
print('The original dictionary is : ',dict)
ascending = sorted(dict.items(), key=operator.itemgetter(0))
print('Dictionary in ascending order by value : ',ascending)
descending = sorted(dict.items(), key=operator.itemgetter(0), reverse=True)
print('Dictionary in descending order by value : ',descending)
```

Ln: 11 Col:

Output

```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Enter the number: 5
[1, 2, 3]
Ln:14 Col:4
```

- **Program 4(a) :** Write a program that takes two lists and returns True if they have at least one Common member.

```
Commonmember.txt - C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32/Com...
File Edit Format Run Options Window Help
def common_element(list1, list2):
    result = False
    for i in list1:
        for j in list2:
            if i == j:
                result = True
                return result
print(common_element([12,25,33,47,50], [50,60,70,80,90]))
print(common_element([11,22,33,34,55], [66,77,88,99]))
Ln:16 Col:0
```

Output

```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
True
None
Ln:34 Col:4
```

- **Program 4(b) :** Write a Python program to print a specified list after removing the 0th, 2nd, 4th and 5th elements.

```
specified position in list.txt - C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32...
File Edit Format Run Options Window Help
list1 = [10,15,20,25,30,35,40,45]
print("Elements after removing 0th,2nd,4th and 5th element from the list")
list1 = [x for (i,x) in enumerate(list1) if i not in (0,2,4,5)]
print(list1)
Ln:8 Col:0
```

Output

```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Elements after removing 0th,2nd,4th and 5th element from the list
[15, 25, 40, 45]
>>>
Ln:41 Col:4
```

- **Program 4(c) :** Write a Python program to clone or copy a list

```
clonelist.txt - C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32/clone...
File Edit Format Run Options Window Help
list1 = [10, 20, 30, 40, 50, 60, 70, 80]
clonelist = list(list1)
print(list1)
print(clonelist)
Ln:9 Col:0
```

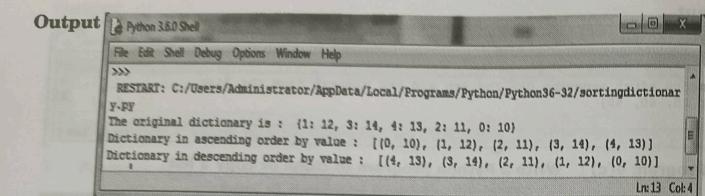
Output

```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
[10, 20, 30, 40, 50, 60, 70, 80]
[10, 20, 30, 40, 50, 60, 70, 80]
Ln:18 Col:4
```

- **Program 5(a) :** Write a Python script to sort (ascending and descending) a dictionary by value.

```
sortingdictionary.txt - C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32/sortin...
File Edit Format Run Options Window Help
import operator
dict = {1: 12, 3: 14, 4: 13, 2: 11, 0: 10}
print('The original dictionary is : ',dict)
ascending = sorted(dict.items(), key=operator.itemgetter(0))
print('Dictionary in ascending order by value : ',ascending)
descending = sorted(dict.items(), key=operator.itemgetter(0), reverse=True)
print('Dictionary in descending order by value : ',descending)
Ln:11 Col:0
```

Output



```

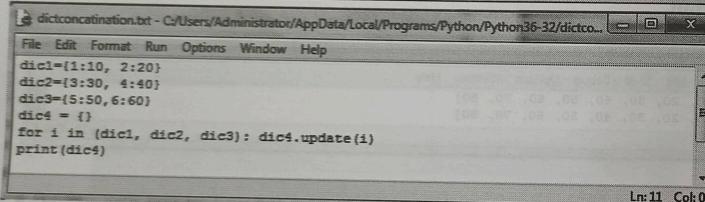
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
>>>
RESTART: C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32/sortingdictionary.py
The original dictionary is : {1: 12, 3: 14, 4: 13, 2: 11, 0: 10}
Dictionary in ascending order by value : [(0, 10), (1, 12), (2, 11), (3, 14), (4, 13)]
Dictionary in descending order by value : [(4, 13), (3, 14), (2, 11), (1, 12), (0, 10)]
  
```

- **Program 5(b) :** Write a Python script to concatenate following dictionaries to create a new one.

Sample Dictionary :

```

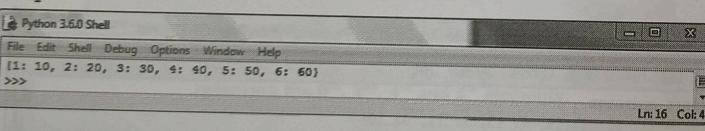
dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}
Expected Result : [1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60]
  
```



```

dictconcatination.txt - C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32/dictco...
File Edit Format Run Options Window Help
dic1=(1:10, 2:20)
dic2=(3:30, 4:40)
dic3=(5:50,6:60)
dic4 = {}
for i in (dic1, dic2, dic3): dic4.update(i)
print(dic4)
  
```

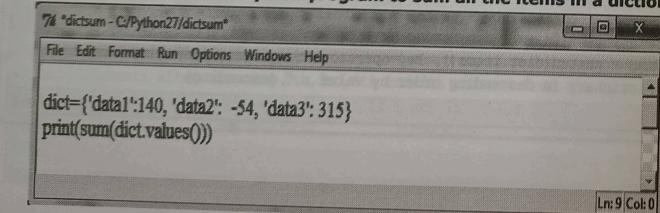
Output



```

Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
[1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60]
  
```

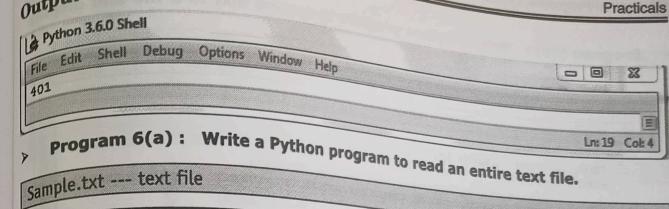
- **Program 5(c) :** Write a Python program to sum all the items in a dictionary.



```

% dictsum - C:/Python27/dictsum*
File Edit Format Run Options Windows Help
dict={'data1':140, 'data2': -54, 'data3': 315}
print(sum(dict.values()))
  
```

Output

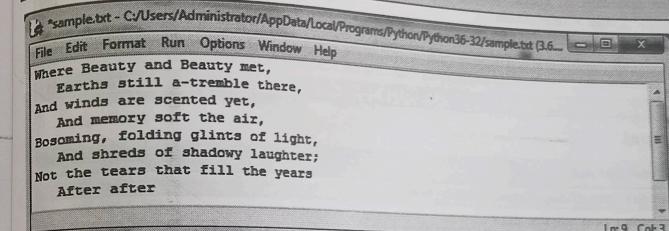


```

Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
401
  
```

➤ **Program 6(a) :** Write a Python program to read an entire text file.

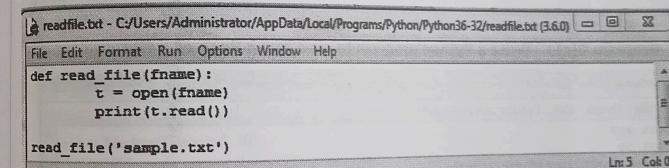
Sample.txt --- text file



```

sample.txt - C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32/sample.txt (3.6.0)
File Edit Format Run Options Window Help
Where Beauty and Beauty met,
Earths still a-tremble there,
And winds are scented yet,
And memory soft the air,
Bosoming, folding glints of light,
And shreds of shadowy laughter;
Not the tears that fill the years
After after
  
```

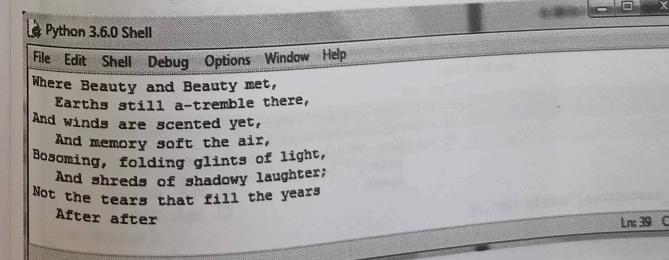
Program



```

readfile.txt - C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32/readfile.txt (3.6.0)
File Edit Format Run Options Window Help
def read_file(fname):
    t = open(fname)
    print(t.read())
read_file('sample.txt')
  
```

Output



```

Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Where Beauty and Beauty met,
Earths still a-tremble there,
And winds are scented yet,
And memory soft the air,
Bosoming, folding glints of light,
And shreds of shadowy laughter;
Not the tears that fill the years
After after
  
```

Program 6(b) : Write a Python program to append text to a file and display the text.

```
appendfile.txt - C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32/appendfile.txt
File Edit Format Run Options Window Help
def read_file(fname):
    from itertools import islice
    with open(fname, "w") as myfile:
        myfile.write("Appended text in the file\n")
        myfile.write("June 2017")
    txt = open(fname)
    print(txt.read())
read_file('sample.txt')

Ln: 4 Col: 55
```

Output

```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Appended text in the file
June 2017
>>>

Ln: 47 Col: 4
```

Program 6(c) : Write a Python program to read last n lines of a file.

```
"readlastnlines.txt" - C:/Users/Administrator/AppData/Local/Programs/Python/Python36-32/readlastnlines.txt
File Edit Format Run Options Window Help
import sys
import os
def read_lastnlines(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-buflen*iter)
            data.extend(f.readlines())
            if len(data) >= lines or f.tell() == 0:
                print(''.join(data[-lines:]))
                break
read_lastnlines('sample.txt', 2)

Ln: 21 Col: 0
```

Output

```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Appended text in the file
June 2017
>>>

Ln: 82 Col: 4
```

Program 7(a) : Design a class that store the information of student and display the same

```
Python 3.4.0: student.txt - C:/Python32/student.txt
File Edit Format Run Options Windows Help
class Student:
    'Common base class for all employees'
    StudentCount = 0

    def __init__(self, name, age):
        self.name = name
        self.age = age
        Student.StudentCount += 1

    def displayCount(self):
        print("Total Student %d" % Student.StudentCount)

    def displayStudent(self):
        print("Name : ", self.name, " , Age: ", self.age)

#This would create first object of Employee class
stud1 = Student("Suvarna", 20)
#This would create second object of Employee class
stud2 = Student("Yogesh", 17)
stud1.displayStudent()
stud2.displayStudent()
print("Total Student %d" % Student.StudentCount)

Ln: 23 Col: 0
```

Output

```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
Name : Suvarna , Age: 20
Name : Yogesh , Age: 17
Total Student 2
>>>

Ln: 49 Col: 4
```

Program 7(b) : Implement the concept of inheritance using python

```
Python 3.4.0: inheritance.txt - C:/Python32/inheritance.txt
File Edit Format Run Options Windows Help
class Polygon:
    def __init__(self, no_of_sides):
        self.n = no_of_sides
        self.sides = [0 for i in range(no_of_sides)]

    def inputSides(self):
        self.sides = [float(input("Enter side "+str(i+1)+" : ")) for i in range(self.n)]

    def dispSides(self):
        for i in range(self.n):
            print("Side", i+1, "is", self.sides[i])

class Triangle(Polygon):
    def __init__(self):
        Polygon.__init__(self, 3)

    def findArea(self):
        a, b, c = self.sides
        # calculate the semi-perimeter
        s = (a + b + c) / 2
        area = (s*(s-a)*(s-b)*(s-c)) ** 0.5
        print("The area of the triangle is %0.2f" % area)

print("L: 24 Col: 0")
```

Output

```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
>>>
>>> t=Triangle()
>>> t.inputSides()
Enter side 1 : 8
Enter side 2 : 9
Enter side 3 : 5
>>> t.dispSides()
Side 1 is 8.0
Side 2 is 9.0
Side 3 is 5.0
>>> t.findArea()
The area of the triangle is 19.90
print("L: 85 Col: 4")
```

- Program 7(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).
- Write a method called add which returns the sum of the attributes x and y.
 - Write a class method called multiply, which takes a single number parameter a and returns the product of a and MULTIPLIER.
 - Write a static method called subtract, which takes two number parameters, b and c, and returns b - c.
 - 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.

```
Python 3.4.0: multiplier.txt - C:/Python32/multiplier.txt
File Edit Format Run Options Windows Help
class Numbers:
    MULTIPLIER = 3.5

    def __init__(self, x, y):
        self.x = x
        self.y = y

    def add(self):
        return self.x + self.y

    @classmethod
    def multiply(cls, a):
        return cls.MULTIPLIER * a

    @staticmethod
    def subtract(b, c):
        return b - c

    @property
    def value(self):
        return (self.x, self.y)

    @value.setter
    def value(self, xy_tuple):
        self.x, self.y = xy_tuple

    @value.deleter
    def value(self):
        del self.x
        del self.y
print("L: 30 Col: 18")
```