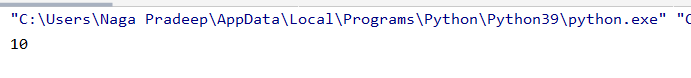
1.1 Write a Python Program to implement your own myreduce() function which works exactly like Python's built-in function reduce()

**Ans: from** functools **import** reduce  
  
**def** myreduce(list):  
 **return** reduce(**lambda** a,b:a+b,list)  
  
mylist=[1,2,3,4]  
  
print(myreduce(mylist))

**Output:**

****

1.2 Write a Python program to implement your own myfilter() function which works exactly like Python's built-in function filter()

**Ans: def** myfilter(lis):  
 **return** list(filter(**lambda** a:a%2==0,lis))  
  
 mylist=[1,2,3,4,5,6,7,8]  
  
 print(myfilter(mylist))

**Output:**

****

**2. Implement List comprehensions to produce the following lists.**

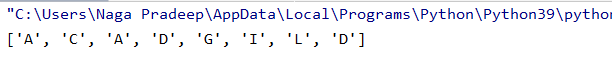
**Write List comprehensions to produce the following Lists**

**['A', 'C', 'A', 'D', 'G', 'I', ’L’, ‘ D’]**

**Ans.word= ”ACADGILD”**

print([i **for** i **in word**])

**Output:**

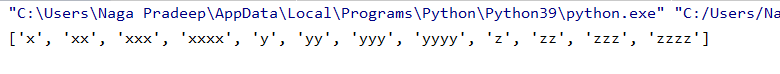


**['x', 'xx', 'xxx', 'xxxx', 'y', 'yy', 'yyy', 'yyyy', 'z', 'zz', 'zzz', 'zzzz']**

**Ans. Word=[“x”,”y”,”z”]**

print([item\*i **for** item **in Word for i in range(1,5)]**)

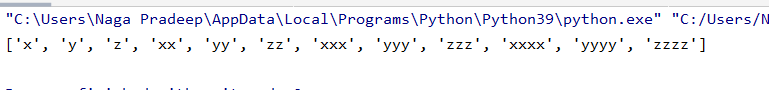
**Output:**

****

**['x', 'y', 'z', 'xx', 'yy', 'zz', 'xx', 'yy', 'zz', 'xxxx', 'yyyy', 'zzzz']**

**Ans.** Word=[**'x'**,**'y'**,**'z'**]  
print([item\*i **for** i **in** range(1,5) **for** item **in** Word])

**Output:**

****

**[[2], [3], [4], [3], [4], [5], [4], [5], [6]]**

**Ans.** num=[2,3,4]  
  
 print([[item+i] **for** item **in** num **for** i **in** range(0,3)])

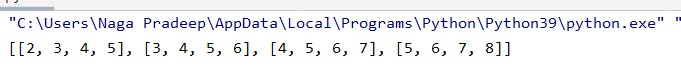
**Output:**



**[[2, 3, 4, 5], [3, 4, 5, 6], [4, 5, 6, 7], [5, 6, 7, 8]]**

**Ans.** lis = [2,3,4,5]  
 print([ [item+i **for** item **in** lis] **for** i **in** range(0,4) ])

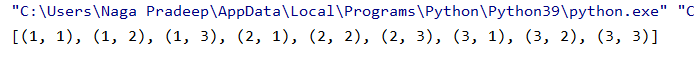
**Output:**



**[(1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (3, 3)**

**Ans.** p=[1,2,3]  
 print([(item,i) **for** item **in** p **for** i **in** range(1,4)])

**Output:**

****

3. Implement a function longestWord() that takes a list of words and returns the longest one.

Ans.

**Task 2:**

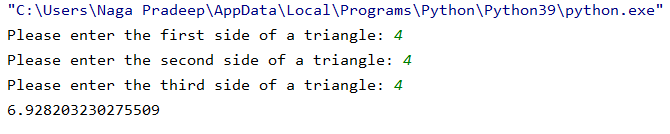
1.1 Write a Python Program(with class concepts) to find the area of the triangle using the below formula.

area = (s\*(s-a)\*(s-b)\*(s-c)) \*\* 0.5

Function to take the length of the sides of triangle from user should be defined in the parent class and function to calculate the area should be defined in subclass.

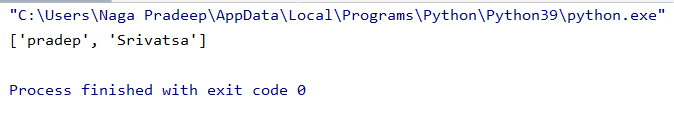
**Ans. import** math  
  
a = int(input(**'Please enter the first side of a triangle: '**))  
b = int(input(**'Please enter the second side of a triangle: '**))  
c = int(input(**'Please enter the third side of a triangle: '**))  
  
  
**class** triangle():  
 **def** \_\_init\_\_(self, a, b, c):  
 self.a = a  
 self.b = b  
 self.c = c  
  
 **def** area(self):  
 s = (a + b + c) / 2  
 area = math.sqrt(s \* (s - a) \* (s - b) \* (s - c))  
 **return** area  
  
  
t = triangle(a, b, c)  
print(t.area())

**Output:**

****

1.2 Write a function filter\_long\_words() that takes a list of words and an integer n and returns the list of words that are longer than n.

**Ans.** lis=[**"pradep"**,**"lop"**,**"Srivatsa"**]  
 n=4  
 D=list(filter(**lambda** a:len(a)>n,lis))  
  
 print(D)

****

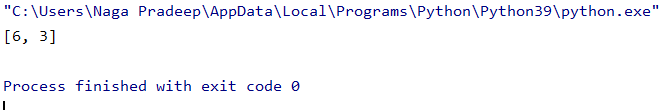
2.1 Write a Python program using function concept that maps list of words into a list of integers representing the lengths of the corresponding words​.

Hint: ​If a list [ ab,cde,erty] is passed on to the python function output should come as [2,3,4]

Here 2,3 and 4 are the lengths of the words in the list.

**Ans.**lis=[**"pradep"**,**"lop"**]  
  
 D=list(map(**lambda** a:len(a),lis))  
  
 print(D)

**Output:**

****

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

**Ans**. **def** vow(n):  
 y=**'aeiou'  
 for** i **in** y:  
 **if** i==n:  
 **return True  
 else**:  
 **return False**print(vow(**'p'**))

**Output:**

