

# WORKSHEET

## PYTHON LIBRARIES & RECURSIONS

1	What is module?
Ans	A module in Python is simply a Python file(.py) which contains functions, class, variables, constants and other elements.
2	What is the file extension of Python module file?
Ans	.py
3	Name any 2 built-in modules and also 2 function belonging to each category/
Ans	Built-in modules: 1) math : sqrt(), pow() 2) statistics : mean(), median()
4	List any two advantages of modules?
Ans	Advantages: a. It reduces program complexity to some degree b. Reusable set of code and thus reduce Line of Code
5	Raj is a Python programmer and creating a project on some statistical application. For some functionality he requires the module <b>statistics</b> . Help Raj to correctly import this module so that all the functionality of statistics will be available in the program.
Ans	import statistics
6	As Raj started coding, he realized he is actually using only single function <b>median()</b> from the module <b>statistics</b> . Now he wants to reduce the load and import only single function from the module statistics. Help Raj to write the correct way to import so that only single function will be imported from statistics.
Ans	from statistics import median
7	Fill in the blanks to import all the name from given module: from statistics import _____
Ans	from statistics import *
8	Raj is a Python programmer and he has imported the module <b>math</b> , and from this math module he wants to use the function sqrt() to calculate square root of n. But he forgot how to use function from imported module. Help Raj to use the function sqrt().  import math n = int(input("Enter number ")) num = _____ # statement to call sqrt() function for value n print(num)
Ans	num = math.sqrt(n)
9	Write the missing statements:  from math import sqrt,pow print(_____) # To calculate square root of 144 print(_____) # To calculate (3) <sup>7</sup> print(math.sqrt(121)) # This statement will work or not? If not, give reason
Ans	print(sqrt(144)) print(pow(3,7))

	<code>print(math.sqrt(121))</code> this statement will not work because we have not imported the math module; we have imported only two function in current program namespace i.e. main, So there is no module name 'math' is available in current program.
10	Give the name of the required module for the given functions? (i) randrange() (ii) mean() (iii) dump() (iv) sin()
Ans	(i) random (ii) statistics (iii) pickle (iv) math
11	_____ function is used to get all information about module i.e. name of all functions, variables, etc available in that module.
Ans	help()
<b>RECURSION</b>	
1	<b>Which is the most appropriate definition for recursion?</b> a) A function that calls itself b) A function execution instance that calls another execution instance of the same function c) A class method that calls another class method d) An in-built method that is automatically called
Ans	<b>a)</b> A function that calls itself
2	To end the recursive function, it must include _____
Ans	Base Condition
3	<b>What happens if the base condition isn't defined in recursive programs?</b> a) Program gets into an infinite loop b) Program runs once c) Program runs n number of times where n is the argument given to the function d) An exception is thrown
Ans	a) Program gets into an infinite loop
4	<b>Which of these is false about recursion?</b> a) Recursive function can be replaced by a non-recursive function b) Recursive functions usually take more memory space than non-recursive function c) Recursive functions run faster than non-recursive function d) Recursion makes programs easier to understand
Ans	<b>c)</b> Recursive functions run faster than non-recursive function
5	<b>Execution of recursive calls in recursion is in which order (Choose correct option)</b> 1. Sequential 2. Reverse
Ans	<b>2. Reverse</b>
6	Define a recursive SCREENSAVER() function which displays "Welcome to my PC" infinite times.
Ans	<pre>def SCREENSAVER():     print("Welcome to my PC")     SCREENSAVER()</pre>
7	<b>Fill in the line of the following Python code for calculating the factorial of a number.</b>

	<pre>def fact(n):     if n==1:         return 1     else:         return _____ a) n * fact(n-1) b) (n-1) * (n-2) c) n * (n-1) d) fact(n) * fact(n-1)</pre>
Ans	<b>a) n * fact(n-1)</b>
8	Define a recursive function FACT(n) to calculate and return the factorial of n
Ans	<pre>def FACT(n):     if n==1:         return 1     else:         return n * FACT(n-1)</pre>
9	Define a recursive function SUM1TON(n) to calculate sum of all the number from 1 to n
Ans	<pre>def sum(n):     if n==1:         return 1     else:         return n+sum(n-1)</pre>
10	Define a recursive function FIBO(n) to generate Fibonacci series for first n numbers For e.g. if n is 6, the numbers to generate are 0 1 1 2 3 5
Ans	<pre>def fibo(n):     if n&lt;=1:         return n     else:         return fibo(n-1)+fibo(n-2)</pre>
11	<p><b>Find the output of following Python code:</b></p> <pre>def Func1(A,B):     if A % B == 0:         return 10     else:         return A + Func1(A,B-1) val = Func1(20,15) print(val)</pre>
Ans	<b>110</b>
12	<p><b>Find the output of following Python code:</b></p> <pre>def Alter(n):     if n&lt;=2:         return 5     else:         return n + Alter(n-2)  sum = Alter(20) print(sum)</pre>

Ans	<b>113</b>
13	<p><b>Find the output of following Python code:</b></p> <pre>def A(n):     if n==1:         return 1     else:         return n + B(n-1) def B(n):     if n==1:         return 5     else:         return n + A(n-1)  val = A(10) print(val)</pre>
Ans	<b>59</b>
14	<p><b>Find the output of following Python code:</b></p> <pre>def fun(x):     if(x &gt; 0):         x -= 1         fun(x)         print(x , end=" ")         x -= 1         fun(x)  a = 4 fun(a)</pre>
Ans	<b>0 1 2 0 3 0 1</b>
15	<p><b>Find the output of following Python code:</b></p> <pre>def fun(n):     if n==4:         return n     else:         return 2*fun(n+1)  x = fun(2) print(x)</pre>
Ans	<b>16</b>
16	<p><b>Find the output of following Python code:</b></p> <pre>def fun(x,y):     if x==0:         return y     else:         return fun(x-1,x+y)</pre>

	<pre>a = fun(4,3) print(a)</pre>
Ans	<b>13</b>
17	<p><b>Find the output of following Python code:</b></p> <pre>def fun(n):     if n==0:         return     print(n%2,end='')     fun(n//2)</pre> <p>fun(25)</p>
Ans	<b>10011</b>
18	<p><b>Which of these is not true about recursion?</b></p> <p>a) Making the code look clean  b) A complex task can be broken into sub-problems  c) Recursive calls take up less memory  d) Sequence generation is easier than a nested iteration</p>
Ans	<b>c) Recursive calls take up less memory</b>
19	<p><b>Which of these is not true about recursion?</b></p> <p>a) It's easier to code some real-world problems using recursion than non-recursive equivalent  b) Recursive functions are easy to debug  c) Recursive calls take up a lot of memory  d) Programs using recursion take longer time than their non-recursive equivalent</p>
Ans	<b>b) ) Recursive functions are easy to debug</b>
20	<p><b>Find the output of the following Python code:</b></p> <pre>def fun1(msg,length):     if length==0:         return     else:         print(msg[length-1],end='')         fun1(msg,length-1)</pre> <p>fun1('amit',4)</p>
Ans	<b>tima</b>