Course Name: B. Tech. (Hons.) CS/ EC (VLSI)

Course Outcome

CO1- Basic Python Programming Concepts
CO2- Working with data types and sorting of data for rich python dataset
CO3- Understanding of looping, control flow statements, the concept of iterators for Python
CO4- Data type conversion and logical analysis for developing core concepts
CO5- Building Concepts in Python string and data collection types (LIST, TUPLE, Dictionary) and its methods
CO6- To learn to write functions and pass arguments in Python. To learn how to build and package Python modules for reusability.

Printed Pages: 04 **University Roll No. .....**

Mid Term Examination, Even Semester 2021-22
B. Tech. (Hons.) CS / EC (VLSI), 1st Year, 2nd Semester
BCSC0053: LEARNING PYTHON FOR DATA ANALYSIS AND VISUALIZATION

Time: 2 Hours **Maximum Marks: 15**

Instruction for students:

1. Read the question paper carefully
2. Understand the ask by reading the explanation
3. Write entire code with proper explanation with comments
4. Proper indentation is mandatory
5. Read the note for each question carefully

Section – A

Attempt All Questions 1 x 3 = 3 Marks

No.	Detail of Question	Marks	CO	BL	KL
1	Is it possible to find the length of a dictionary in Python3? What's the difference between list.pop () and dict.pop()? Explain both in detail with suitable examples.	1	CO2	U	C
2	A chocolate distributor unit has installed two new automatic arms for unloading the chocolate bars from containers. Arm A has the capacity to unload one chocolate bar whilst the other arm B unloads two bars at a time. In order for any two containers to be unloaded fully and simultaneously by both arms, the distributors have to choose the correct chocolate bars quantity (quantity "X" for container unloaded by arm A and quantity "Y" for container unloaded by arm B) in those containers from the supplier. The task to develop a code to identify a pair of container quantities (maximum quantity 5000) such that both	1	CO1	AN	S

	arms unload all chocolate bars from those containers fully and complete their unloading simultaneously so the following containers can be placed for unloading automatically. The correct pair identified can be marked as "Yes" and "No" for incorrect pairs. Example 1: Input: 100—Value of X 200— Value of Y Output: Yes—Print Yes indicating 100 and 200 chocolate bars can be fully emptied simultaneously Explanation: Arm A unloads 100s bars in 100 times Arm B also unloads 200 bars in 100 times, hence both the containers are emptied at the same time and the next pair of containers can be automatically placed for unloading. Hence, the output is a 'Yes'.				2
3	What are the Python modules? Explain Python standard module math and its uses with suitable examples. What is the difference between function round() and function floor()? Explain both with suitable examples.	1	CO6	A	F

Section – B

Attempt All Questions 2 X 3 = 6 Marks

No.	Detail of Question	Marks	CO	BL	KL
1	Write a program to find what value which is needed to be added with a given number (input by the user) to make the result as the next upcoming prime number. Input 35 Output 2 Explanation: The program takes input from the user in form of an int number i.e. say 35. The nearest prime number is 37. So, 2 is needed to be added with 35 so that sum is the next upcoming prime number which is 37.	2	CO4	E	PC

2	Write a function to flatten a nested dictionary. Namespace the keys with a period. For example, given the following dictionary: {"key": 3, "foo": {"a": 5, "bar": {"baz": 8 }}} It should become: {"key": 3, "foo.a": 5, "foo.bar.baz": 8} You can assume keys do not contain dots in them, i.e. no clobbering will Occur	2	CO4	AN	D

3	Write a function expanding (l) that takes input as a list of integers l and returns True if the absolute difference between each adjacent pair of elements strictly increases.
	Here are some examples of how your function should work.
	>>> expanding ([1, 3, 7, 2, 9])
	True
	Explanation: Differences between adjacent elements are 3-1 = 2, 7-3 = 4,
	7-2 = 5 and 9-2 = 7.
	
	>>> expanding ([1, 3, 7, 2,-3])
	False
	Explanation: Differences between adjacent elements are 3-1 = 2, 7-3 = 4,
	7-2 = 5, 2-(-3) = 5, so not strictly increasing.
	>>> expanding ([1, 3, 7, 10])
	False
	Explanation: Differences between adjacent elements are 3-1 = 2, 7-3 = 4,
	10-7 = 3, so not increasing.
		2	CO3	E	D

Section – C

Attempt All Questions 3 x 2 = 6 Marks

No.	Detail of Question	Marks	CO	BL	KL

1	Write a function is_sorted(tp), which accepts the argument tp(tuple of the integers) and returns 0 if tuple elements are arranged in ascending order, and returns 1 if elements are arranged in descending order.
	Return -1 if elements are unsorted.
	def is_sorted(tp: tuple)-> bool:
	# write logic here
	t = eval(input())
	out = is_sorted(t)
	print(out)
	sample test cases
	>>> is_sorted((3, 5, 6, 10))
	0
	>>> is_sorted((2, 8, 5, 7))
	-1
	>>> is_sorted((9, 3, 1, 0))
	1
		3	CO5	U	PC

2	Given a string with some repeated characters, your task is to write a Python script to rearrange the string so that no two adjacent characters are the same. If the given string is successfully rearranged in the correct order (according to the given task), display "True" and if this is not possible, display "False". For example, the given string "aaabbc", can be arranged "ababac" so the output will be "True". For the string "aaab" output should be "False".	3	CO5	U	D