

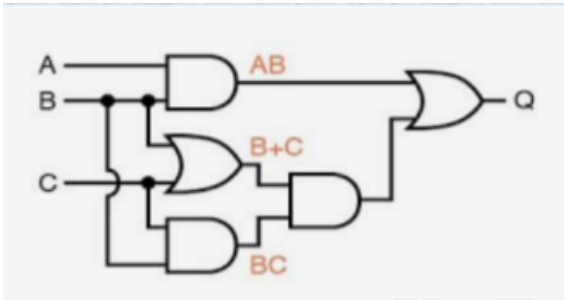
SET-3
KENDRIYA VIDYALAYA SANGATHAN, LUCKNOW REGION
CUMULATIVE EXAMINATION – 2023-24
MARKING SCHEME

Class : XI
Time Allowed: 3 Hours

Subject: Computer Science (083)
Maximum Marks: 70

SECTION - A		
1.	iii) my-name 1 marks for correct answer	1
2.	iii) secondary memory (1 marks for correct answer)	1
3.	in , not (1/2 marks for each correct answer)	1
4	(i) Digital Video Disk (ii) Programmable Read Only Memory (1 marks (½ marks for each correct full form))	1
5.	Windows XP,Wndows10,Linux,Macintosh ½ marks for each correct answer	1
6	ii) 786 1 marks for correct answer	1
7	Bharat 1 marks for correct answer	1
8	Ans : 25 1 marks for correct output	1
9	Ans: 888 1 marks for correct output	1
10	Ans :0 1 marks for correct output	1
11	ii) 1 1 marks for correct output	1
12	ii) F 1 marks for correct output	1
13	i) break	1
14	[["are","a","few","words"]] One marks for correct answer	1
15	ii) Application software 1 marks for correct answer	1
16	Ans : In the computer system, ALU is a main component of the central processing unit, which stands for arithmetic logic unit and performs arithmetic and logic operations. 1 marks for correct answer	1
17	Ans ; (a) (1 mark for correct)	1
18	Ans (d) (1 marks for correct answer)	1
SECTION – B		

19	An infinite loop in Python, or any programming language, is a loop that runs indefinitely and does not have a built-in termination condition. These loops continue executing their code repeatedly without stopping until they are interrupted externally. Example: while True: # This loop will run indefinitely print("This is an infinite loop") (1 mark for Explanation and 1 mark for example)	2
20	• A Variables in Python is only a name given to a memory location, all the operations done on the variable effects that memory location. • A variable in Python is a representational name that serves as a pointer to an object. Example x=10 1 ½ marks for correct explanation and ½ mark for example.	2
21	i) (53)₁₆ ii) (5274)₈ 1 marks for each correct answer	2
22	<pre>city = "mumbai" if city=="jaipur": print("hot weather") elif city == "shimla" : print("very cold ") else: print(city)</pre> Quotes on “Mumbai” == operator in if statements elif in place of elseif else : (½ marks for each correction)	2
23	The break statement in Python is used to exit or terminate a loop prematurely. It is typically used within loops (like for or while loops) to interrupt the normal flow of iteration based on a certain condition . Example: while True: user_input = input("Enter 'q' to quit: ") if user_input == 'q': break # Exit the loop if the user enters 'q' else: print("You didn't quit.") or Type casting in Python refers to the process of converting a value from one data type to another. Python provides built-in functions to perform type casting. This is sometimes necessary when you want to perform operations on values of different data types or when you need to ensure that a value is of a specific data type. examples: X = “ 1234” Y= int(X)	2

	1 marks for correct explanation and 1 marks for correct example.	
24	$A.B + \overline{A.C}$ 2 Marks for Correct Answer , 1 mark for partially correct answer.	2
25	x=20 while x>10: print(x) x= x-3 (½ marks for each correct statement)	2
SECTION – C		
26	 (1 mark for partial correct/term answer if any 1 term is correct)	3
27	In Python, identity operators are used to compare the memory location of two objects to check whether they are the same object. There are two identity operators: is: The is operator is used to check if two variables reference the same object in memory. a=2 b=3 print(a is b) output: false (1 ½ marks for correct explanation) is not: The is not operator is used to check if two variables do not reference the same object in memory. Print (a is not b) Output : true (1 ½ marks for correct explanation)	3
28	Internal Memory (Primary Memory): Internal memory, also known as primary memory or RAM (Random Access Memory), is the computer's main storage area that is directly accessible by the CPU. It is volatile, meaning it loses all data when the computer is turned off or restarted. It is used to store data and instructions that are actively being used or processed by the CPU. It provides fast access to data, which allows the CPU to perform tasks efficiently. Example: RAM modules in a computer. (1 Marks for correct definition, ½ for correct example) External Memory (Secondary Memory):	3

	External memory, also known as secondary memory or storage, refers to devices or media that store data and programs for the long term, even when the computer is powered off. It is non-volatile, meaning it retains data even when the computer is turned off. External memory provides a large and relatively permanent storage space for files, documents, applications, and the operating system. Example: Hard drives, SSDs, USB drives, DVDs, etc (1 Marks for correct definition, ½ for correct example)			
29	<pre>r= int(input(" enter radius of a circle")) area=3.14*r*r peri=2*3.14*r print("area=", area) print(" perimeter=", peri)</pre> (1 mark for correct input , 1 marks for correct output, 1 mark for correct calculation) <i>Any other suitable code producing desired output will also be marked as correct answer</i>	3		
30	<pre>str=input("Enter Any String") for i in range(0, len(str)//2): if str[i] != str[len(str)-i-1]: print("String is Not Palindrome") break else: print("String is Palindrome")</pre> (1 mark for correct input and output, 1 marks for correct loop, 1 marks for correct if statement) Or <pre>str=input("Enter Any String") if str==str[::-1]: print("Palindrome") else: print("Not a Palindrome") </pre> (1 mark for correct input and output, 1 marks for SLICING, 1 marks for correct if statement) <i>Any other suitable code producing desired output will also be marked as correct answer</i>	3		
SECTION – D				
31	A list in Python is a collection of items, which can be of any data type (such as integers, floats, strings, or even other lists). Lists are ordered, mutable (can be changed), and allow duplicate values. <table><tr><td>append()</td><td>Extend()</td></tr></table>	append()	Extend()	5
append()	Extend()			

	<table><tr><td>To add a single entry to the end of a list, we use append() function.</td><td>To add additional elements or an iterable to the end of a list, use the extend() function</td></tr><tr><td>accepts only one input element.</td><td>accepts as input an iterable (such as a list or tuple).</td></tr><tr><td>The append() function adds the full input to the list as a single item.</td><td>extend() adds each item to the list independently after iterating through each one in the input.</td></tr><tr><td>Examples: >>>L=[1,2,3,4,5] >>>L.append(10) >>>L [1, 2, 3, 4, 5, 10] >>>L.append([10,20,30]) >>>L [1, 2, 3, 4, 5, 10, [10, 20, 30]]</td><td>Examples: >>>LL=[1,2,3,4,5] >>>LL.extend([10,20,30]) >>>LL [1, 2, 3, 4, 5, 10, 20, 30]</td></tr></table>	To add a single entry to the end of a list, we use append() function.	To add additional elements or an iterable to the end of a list, use the extend() function	accepts only one input element.	accepts as input an iterable (such as a list or tuple).	The append() function adds the full input to the list as a single item.	extend() adds each item to the list independently after iterating through each one in the input.	Examples: >>>L=[1,2,3,4,5] >>>L.append(10) >>>L [1, 2, 3, 4, 5, 10] >>>L.append([10,20,30]) >>>L [1, 2, 3, 4, 5, 10, [10, 20, 30]]	Examples: >>>LL=[1,2,3,4,5] >>>LL.extend([10,20,30]) >>>LL [1, 2, 3, 4, 5, 10, 20, 30]	
To add a single entry to the end of a list, we use append() function.	To add additional elements or an iterable to the end of a list, use the extend() function									
accepts only one input element.	accepts as input an iterable (such as a list or tuple).									
The append() function adds the full input to the list as a single item.	extend() adds each item to the list independently after iterating through each one in the input.									
Examples: >>>L=[1,2,3,4,5] >>>L.append(10) >>>L [1, 2, 3, 4, 5, 10] >>>L.append([10,20,30]) >>>L [1, 2, 3, 4, 5, 10, [10, 20, 30]]	Examples: >>>LL=[1,2,3,4,5] >>>LL.extend([10,20,30]) >>>LL [1, 2, 3, 4, 5, 10, 20, 30]									
	(2 marks for List Explanation+ 2 marks for at least two differences marks+ 1 mark for exmaples)									
32	<pre>n = 5 # The number of rows for i in range(n, 0, -1): for j in range(i, 0, -1): print(j, end="") print() # Move to the next line after each row</pre> (2 marks for correct outer loop, 2 marks for correct inner loop, 1 mark for correct print statements)	5								
33	<pre># Input the first time in hours and minutes hours1 = int(input("Enter the first time (hours): ")) minutes1 = int(input("Enter the first time (minutes): ")) # Input the second time in hours and minutes hours2 = int(input("Enter the second time (hours): ")) minutes2 = int(input("Enter the second time (minutes): ")) # Calculate the total hours and minutes total_hours = hours1 + hours2 total_minutes = minutes1 + minutes2 # Adjust for extra minutes if total_minutes >= 60: total_hours += total_minutes // 60 total_minutes %= 60 # Display the total time print("Total Time:", total_hours ,"::" total_minutes")</pre> (2 marks for input , 2 marks for calculation , 1 marks for output)	5								
SECTION - E										
34	DeMorgan's First Theorem (X.Y)' = X' + Y'	4								

	<table><tr><th colspan="2">Input Variables</th><th colspan="5">Output Terms</th></tr><tr><th>X</th><th>Y</th><th>X.Y</th><th>(X.Y)'</th><th>X'</th><th>Y'</th><th>X'+Y'</th></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> DeMorgan's Second theorem: $(X+Y)' = X'.Y'$ Truth Table: <table><tr><th colspan="2">Input Variables</th><th colspan="5">Output Terms</th></tr><tr><th>X</th><th>Y</th><th>X+Y</th><th>(X+Y)'</th><th>X'</th><th>Y'</th><th>X'.Y'</th></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> (2 marks for each correct answer)	Input Variables		Output Terms					X	Y	X.Y	(X.Y)'	X'	Y'	X'+Y'	0	0	0	1	1	1	1	1	0	0	1	0	1	1	0	1	0	1	1	0	1	1	1	1	0	0	0	0	Input Variables		Output Terms					X	Y	X+Y	(X+Y)'	X'	Y'	X'.Y'	0	0	0	1	1	1	1	1	0	1	0	0	1	0	0	1	1	0	1	0	0	1	1	1	0	0	0	0	
Input Variables		Output Terms																																																																																				
X	Y	X.Y	(X.Y)'	X'	Y'	X'+Y'																																																																																
0	0	0	1	1	1	1																																																																																
1	0	0	1	0	1	1																																																																																
0	1	0	1	1	0	1																																																																																
1	1	1	0	0	0	0																																																																																
Input Variables		Output Terms																																																																																				
X	Y	X+Y	(X+Y)'	X'	Y'	X'.Y'																																																																																
0	0	0	1	1	1	1																																																																																
1	0	1	0	0	1	0																																																																																
0	1	1	0	1	0	0																																																																																
1	1	1	0	0	0	0																																																																																
35	i) Good morning#Good Morning ii) [1, 3, 5, 7] (2 marks for each correct output)	4																																																																																				
