967, 968
C : 112

Air University

(Mid-Term Examination: Fall 2024) 237265

Subject:	**Data Structures**	Total Marks:	**50**
Course Code:	**CS-214**	Date:	
Class:	**BS-CYS**	Time:	
Semester:	**III**	Duration:	**2 Hours**
Section:	**A/B**	FM Name:	**Ms. Mehmoona Jabeen**

HoD Signatures:_____ FM Signatures:_____

Note:

- **All questions must be attempted.**
- **This examination carries 25% weight towards the final grade.**
- **Return the question paper with the answer sheet**

	Q. No. 1 (CLO 1)	**16 Marks**
a	**Consider a queue implemented using an array. Both enqueue (inserting an element) and dequeue (removing an element) operations currently take O(1) time. However, managing the 'front' and 'back' pointers raises important questions.** i. If the 'back' pointer reaches the end of the array, what potential problem could arise? Suggest a solution to handle this situation without increasing time complexity. ii. What happens if the 'front' pointer moves ahead of the 'back' pointer? In what scenario could this occur, and how would you detect it? iii. What does the condition 'front == back' indicate? Does it mean the queue is full, empty, or something else? Justify your answer with an example.	6
b	For each of the following scenarios, choose the "best" data structure from the following list: an Array, Linked List, Doubly Linked List, Circular Linked List, Stack, Queue. In each case, briefly justify your answer. i. A printer management system needs to manage print jobs in the order they are received, ensuring that the first job submitted is the first to be printed. ii. A text editor needs to provide the functionality to undo the last series of edits made by the user. iii. A traffic light control system needs to manage a sequence of traffic signals where the light changes in a continuous cycle, and the system must always move to the next signal in the sequence.	6
c	You are working on a data streaming system where incoming data packets are temporarily stored in a queue. This system needs to handle a fixed amount of memory efficiently, meaning the size of the buffer is limited. If the buffer becomes full and new data arrives, the oldest data at the front of the queue will be overwritten by the new one. You need to provide the solution to this problem along with the complete algorithm.	4
	Q. No. 2 (CLO 2)	**16 Marks**
a	Assume a function **LoginProcess** is executing, and it calls another function **AuthenticateUser.** The **LoginProcess** function needs to store its local variables in memory so that they can be restored when **AuthenticateUser** completes. Now, **AuthenticateUser** calls another function **VerifyOTP.** The local variables of AuthenticateUser must also be stored in memory so that it can resume execution once **VerifyOTP** completes.	3+3

	(i) Which data structure is suitable for storing local variables in memory for this scenario? (ii) What operations will be required in a data structure?	
b	You are required to manage function calls in an application using a tree structure, where each node represents a function call, storing its local variables. The root node represents the LoginProcess function, which calls AuthenticateUser, and then VerifyOTP. you would implement a tree where each node contains local variables and a reference to its parent function call. **Operations: Implement the following:** AddFunctionCall: Add a new function call as a child, RemoveFunctionCall: Remove a completed function call,	10
	Q. No. 3 (CLO 3)	**18 Marks**

a	**i. Calculate the time complexity of the following code.**
	int x = 0, y = 0;
	for (int p = 0; p < n; p++) {
	if (p < m) {
	x = p;
	y = n - p;
	}
	else {
	for (int q = m; q < n; q++) {
	x = x + q;
	y = x - m;
	}
	}
	}
		9

b	i. How does the use of a circular queue solve the issue of 'back' reaching the end of the array? Does the time complexity of enqueue and dequeue operations remain O(1) with this solution? Why or why not? ii. In a Singly Linked List (SLL), removing the last node (tail) requires O(n) time complexity because you need to traverse the entire list to find the second-last node.What strategies or structural modifications can be employed to reduce the time complexity of deleting the last node from a list to O(1)? Please explain your approach in detail. iii. Find the time complexity of both functions used in the Q 02 (a).	3+3+3

********** End of Question Paper **********