SET B – ANSWER KEY

SRM Institute of Science and Technology
College of Engineering and Technology
School of Computing

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

Academic Year: 2023-24 (ODD)

Test: CLA-CT-1 **Date: 04-09-2023**
Course Code & Title: 21CSC201J Data Structures and Algorithms **Duration:** 1 hour 40 min
Year & Sem: II Year / III Sem **Max. Marks:** 50

Course Articulation Matrix:

Course Outcome	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Program Specific Outcomes		
													PSO-1	PSO-2	PSO-3
CO1	1	2	3	-	-	-	-	-	-	-	-	3	3	-	-
CO2	2	3	3	-	-	-	-	-	-	-	-	3	3	-	-
CO3	2	3	3	-	-	-	-	-	-	-	-	3	3	-	-
CO4	2	3	3	-	-	-	-	-	-	-	-	3	3	-	-
CO5	3	2	3	-	-	-	-	-	-	-	-	3	3	-	-

Part - A
(10 x 1 = 10 Marks)

Instructions: Answer all

Q. No	Question	Marks	BL	CO	PO	PI Code
1	A data structure that can store related information together is called (a) Array (b) String (c) Structure (d) All of these	1	L1	1	2	2.5.1
2	Which notation comprises a set of all functions h(n) that are greater than or equal to cg(n) for all values of n ≥ n0 ? (a) Omega notation (b) Big O notation (c) Small o notation (d) Theta Notation	1	L1	1	1	1.7.1

3	What will be the output for the following C code?
	#include<stdio.h>
	void main() {
	  int a[] = {8,7,3,2,4}, *p;
	  p = a;
	  ++*p;
	  printf("%d ", *p);
	p += 2;
	  printf("%d ", *p);
	}
	a) 2 4
	b) 3 4
	c) 9 3
		1	L2	1	1	1.7.1

	d) 2 3					

4	How do you access Employee's salary for the given structure?
	
	struct Employee
	{
	  int employee_id;
	  char name[20];
	  int salary;
	};
	struct Organisation
	{
	  char organisation_name[20];
	  char org_number[20];
	  struct Employee emp;
	}org;
	
	a) salary.org.emp
	b) org.emp.salary
	c) salary.emp.org
	d) emp.org.salary
		1	**L1**	**1**	**2**	**2.5.2**

5	What is the function of realloc()? (a) Change the location of memory allocated by malloc() or calloc(). (b) Reallocates memory deleted by free() function. (c) It is used to modify the size of the previously allocated memory space. (d) None of these	**1**	**L1**	**1**	**2**	**2.5.1**
6	Identify the option which is NOT true about an array? a. A zero index is used to refer to the first element b. It understands the concept of ordered elements. c. Duplicate values are allowed d. Expand automatically when an array is full.	**1**	**L1**	**2**	**2**	**2.5.1**
7	Which of the following information is stored in a doubly-linked list's nodes? a. Value of node b. Address of next node c. Address of the previous node d. All of the above	**1**	**L1**	**2**	**2**	**2.5.3**
8	In a Circular linked list organization, insertion of a record involves the modification of a. No pointer b. 1 pointer c. 2 pointer d. 3 pointer	**1**	**L1**	**2**	**2**	**2.5.1**

9	Polynomial addition can be implemented using which of the following data structures? a) Linked List b) Trees c) Stacks d) Priority Queue	**1**	**L2**	**2**	**1**	**1.7.1**
10	Which of the following is a practical example of a doubly linked list? a) A browser cookie file. b) A quest in a game that lets users retry stages. c) A game in which the player runs forward. d) A first-in-first out scheduling system.	**1**	**L3**	**2**	**1**	**1.7.1**

Part – B

(4 x 5 = 20 Marks)

Instructions: Answer ALL

11	Create pointers to structures named dog. Input dogname, breed, age, color. Print the details of the dog.
	Mandatory condition: use only . and *. Do not use ->
	Answer:
	#include <stdio.h>
	#include <stdlib.h>
	
	// Define the dog structure
	struct Dog {
	    char name[50];
	    char breed[50];
	    int age;
	    char color[50];
	};
	int main() {
	    // Declare a dog structure
	    struct Dog myDog;
	    // Input dog details using pointers and dot (.)
	    printf("Enter dog's name: ");
	    scanf("%s", myDog.name);
	    printf("Enter dog's breed: ");
	    scanf("%s", myDog.breed);
	    printf("Enter dog's age: ");
	    scanf("%d", &myDog.age);
		5	**L3**	**1**	**1**	**1.7.1**

	printf("Enter dog's color: ");
	scanf("%s", myDog.color);
	
	// Print the details of the dog using pointers and dot (.)
	printf("\nDog's Details:\n");
	printf("Name: %s\n", myDog.name);
	printf("Breed: %s\n", myDog.breed);
	printf("Age: %d\n", myDog.age);
	printf("Color: %s\n", myDog.color);
	
	return 0;
	}					

12	Write a self-referential structure and implement any one of the dynamic memory allocation techniques on the self-referential structure.
	--- 	
	// Define a self-referential structure for a singly linked list node
	struct Node {
	int data;
	struct Node *next; // Pointer to the next node in the list
	};
	
	int main() {
	// Create nodes dynamically and build a linked list
	struct Node *head = NULL; // Initialize the list as empty
	
	// Add nodes to the linked list
	for (int i = 1; i <= 5; i++) {
	// Create a new node
	struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
	
	// Check if memory allocation was successful
	if (newNode == NULL) {
	printf("Memory allocation failed.\n");
	return 1; // Exit with an error code
	}
	
	// Assign data to the new node
	newNode->data = i;
	
	// Make the new node point to the current head of the list
		5	**L3**	**1**	**2**	**2.6.3**

	newNode->next = head;
	
	        // Update the head to point to the new node, making it the new head
	        head = newNode;
	    }
	
	    // Traverse and print the linked list
	    printf("Linked List: ");
	    struct Node *current = head;
	    while (current != NULL) {
	        printf("%d -> ", current->data);
	        current = current->next;
	    }
	
	    return 0;
	}					

13	Write a program to print the total number of occurrences of a given item in the linked list.
	
	// Function to count the occurrences of a given item in the linked list
	int countOccurrences(struct Node *head, int item)
	{
	    int count = 0;
	    struct Node *current = head;
	
	    while (current != NULL) {
	        if (current->data == item) {
	            count++;
	        }
	        current = current->next;
	    }
	
	    return count;
		5	**L3**	**2**	**2**	**2.5.2**

14	The fruit vendor, Anwar, wants to count every apple in the basket. We must create software that can handle n numbers of apples and display the count for each number because the apples are numbered. Write the appropriate pseudocode.
	#include <stdio.h>
	
	int main() {
	    int n, i;
	// Input the total number of apples in the basket
	    printf("Enter the total number of apples: ");
	    scanf("%d", &n);
	
	    // Declare an array to store counts for each apple number
	    int appleCounts[1000] = {0}; // Assuming apple numbers are between 1 and 1000
	
	    // Input and count each apple
		5	**L3**	**2**	**2**	**2.5.2**

	for (i = 0; i < n; i++) {
	int appleNumber;
	printf("Enter the apple number %d: ", i + 1);
	scanf("%d", &appleNumber);
	
	// Increment the count for the corresponding apple number in the array
	appleCounts[appleNumber]++;
	}
	
	// Display the count for each apple number
	printf("Apple Counts:\n");
	for (i = 1; i <= 1000; i++) { // Assuming apple numbers are between 1 and 1000
	if (appleCounts[i] > 0) {
	printf("Apple number %d: Count %d\n", i, appleCounts[i]);
	}
	}
	
	return 0;
	}					

Part – C

(2 x 10 = 20 Marks)

Instructions: Answer TWO Questions

15	Consider a goods train, which initially has an engine as a header. Later compartments are linked with engines one by one in series. The compartments can be attached or detached with the engine or in between. Implement a data structure for the following operations in the goods train 1. Attaching a compartment with the engine **Ans:** Step 1: IF AVAIL = NULL Write OVERFLOW Go to Step 12 [END OF IF] Step 2: SET = AVAIL Step 3: SET AVAIL = AVAIL NEXT Step 4: SET DATA = VAL Step 5: SET PTR = START Step 6: SET PREPTR = PTR Step 7: Repeat Steps 8 and 9 while != NUM Step 8: SET PREPTR = PTR Step 9: SET PTR = PTR NEXT [END OF LOOP] Step 1 : PREPTR NEXT = Step 11: SET NEW_NODE NEXT = PTR Step 12: EXIT 2. Detaching a compartment from the engine **Ans** Step 1: IF START = NULL	**10**	**L3**	**1**	**1**	**1.7.1**

	Write UNDERFLOW Go to Step 1 [END OF IF] Step 2: SET PTR = START Step 3: SET PREPTR = PTR Step 4: Repeat Steps 5 and 6 while PREPTR DATA! = NUM Step 5: SET PREPTR = PTR Step 6: SET PTR = PTR NEXT [END OF LOOP] Step 7: SET TEMP = PTR Step 8: SET PREPTR NEXT = PTR NEXT Step 9: FREE TEMP Step 1 : EXIT					
(or)						
16.	a) Find the tight bound of running time of a cubic function. f(n) = 2n3+3n+6; g(n) = n3 $f(n) = \theta(g(n)) = O(n^3)$ for $c_1=2$, $c_2 = 11$, $n_0 = 1$ (with proper derivation) – **5 Marks** b) Prove that 10n3 + 20n ≠ O(n2) Since f(n) is greater than g(n), it is not possible to find tightest upper bound using the given g(n). If the proper justification is given with examples **5 marks** can be awarded.	**5+5**	**L3**	**1**	**1**	**1.7.1**

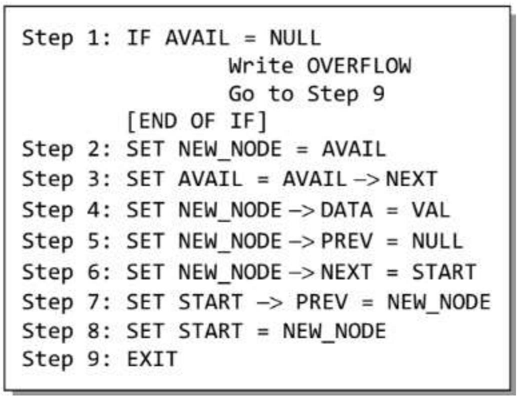
17	Consider a data structure with the elements 22,32,44,51,65,71,80 which are stored in adjacent memory locations. The element 44 needs to be deleted first and after that you want to insert the data element 92 at the 4th position. Write the appropriate pseudo code to perform the aforementioned operations and provide a pictorial representation of it.
	
	#include <stdio.h>
	
	int main() {
	
	int data[] = {22, 32, 44, 51, 65, 71, 80};
	
	int n = sizeof(data) / sizeof(data[0]);
	
	// Pictorial representation before deletion and insertion
	
	printf("Before:\n");
		10	**L3**	**2**	**2**	**2.5.2**

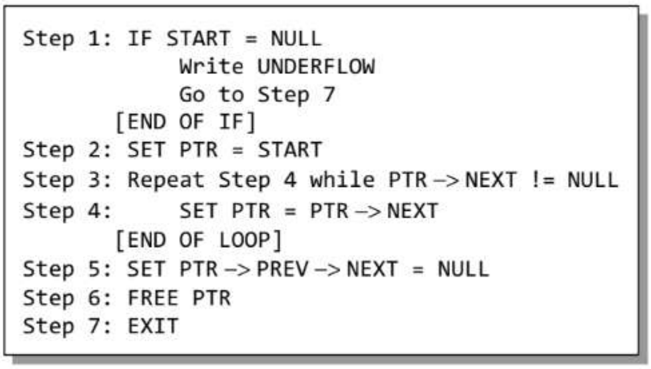
```
    for (int i = 0; i < n; i++) {
        printf("%d ", data[i]);
    }
    printf("\n");

    // Deleting element 44 (3rd position)
    int deleteIndex = 2; // Index of element to be deleted
    for (int i = deleteIndex; i < n - 1; i++) {
        data[i] = data[i + 1];
    }
    n--; // Reduce the size of the array

    // Inserting element 92 at the 4th position (3rd index)
    int insertIndex = 3; // Index where element 92 will be inserted
    for (int i = n; i > insertIndex; i--) {
        data[i] = data[i - 1];
    }
    data[insertIndex] = 92;
    n++; // Increase the size of the array

    // Pictorial representation after deletion and insertion
    printf("After:\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", data[i]);
    }
    printf("\n");
```

	return 0; } output: Data: 22 32 44 51 65 71 80 After Deletion of 44 and Insertion of 92: Data: 22 32 92 51 65 71					
(or)						
18	David prepared the list of items needed for his college tour based on its priority. He added 6 items (Item A, Item B, Item C, Item D, Item E and Item F). in the list based on its priority in the list. After that, he realized that the second item had a higher priority than the first one for his tour trip. So he is trying to reorder things. And later on he realized that the last item from the list he prepared is no more required. Try to help him to reorder the list using a doubly linked list with proper pseudocode and give a visual representation of it. Insertion at beginning:  Step 1: IF AVAIL = NULL Write OVERFLOW Go to Step 9 [END OF IF] Step 2: SET NEW_NODE = AVAIL Step 3: SET AVAIL = AVAIL -> NEXT Step 4: SET NEW_NODE -> DATA = VAL Step 5: SET NEW_NODE -> PREV = NULL Step 6: SET NEW_NODE -> NEXT = START Step 7: SET START -> PREV = NEW_NODE Step 8: SET START = NEW_NODE Step 9: EXIT Deletion at the end:	**10**	**L3**	**2**	**2**	**2.5.2**

<table>
<tr>
<td></td>
<td>
<pre>
Step 1: IF START = NULL
            Write UNDERFLOW
            Go to Step 7
       [END OF IF]
Step 2: SET PTR = START
Step 3: Repeat Step 4 while PTR -> NEXT != NULL
Step 4:   SET PTR = PTR -> NEXT
       [END OF LOOP]
Step 5: SET PTR -> PREV -> NEXT = NULL
Step 6: FREE PTR
Step 7: EXIT
</pre></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</table>