



RV College of Engineering®  
Mysuru Road, RV Vidyanikethan Post,  
Bengaluru - 560059, Karnataka, India

Question Paper ID: 128962

Course Code: IS233AI

USN 

1	R	V	2	4	1	5	0	0	5
---	---	---	---	---	---	---	---	---	---

RV College of Engineering

(An Autonomous Institution Affiliated to VTU)

R V Vidyanikethan Post  
Mysuru Road Bengaluru - 560 059

III Semester B.E. Regular / Supplementary Examination February/March 2026

COMMON TO AIML / CD / CS / CY / IS

Data Structures and Applications

Time : 3 Hours

Maximum Marks : 100

**Instructions to the students**

1. Answer all questions from Part A . Part A questions should be answered in first three pages of the answer book only.
2. Answer Five full questions from Part B. In Part B question number 2 is compulsory. Answer any one full question from 3 and 4, 5 and 6, 7 and 8, 9 and 10.

**Part A**

Question No	Question	M	CO	BT
1.1	Convert the given infix expression to postfix expression directly without using stack. $a + b * (c^d - e) ^ (f + g * h) - i$	02	2	3
1.2	Which data structure is best suited for storing employee records when frequent insertion and deletion operations are required, and why?	02	1	3
1.3	A circular queue is implemented using an array of size 10. The array index starts with 0, front is 6, and rear is 9. The insertion of next element takes place at the array index_____.	02	1	1
1.4	Analyze the function to insert at the beginning in a Singly Linked List. Identify any errors or write the expected behavior. void insertAtBeginning(struct Node *head, int val){ struct Node *newNode = (struct Node*)malloc(sizeof(struct Node)); newNode->data = val; newNode->next = head; head = newNode; }	02	1	5
1.5	In a circular singly linked list, what will the following code snippet do if head points to the first node and the list is not empty?  struct Node { int data; struct Node* next; };	02	1	3

```
// ... inside a function
```

```
struct Node* temp = head;
```

```
while (temp->next != head) {
```

```
    temp = temp->next;
```

```
}
```

1.6	Which linked list is best suited for implementing a bank queue that operates continuously?	01	2	2
1.7	Find the maximum number of nodes in a skew tree of height h. Assume that root node is at level 0	01	1	1
1.8	In a max-heap, which operation is performed to restore the heap property after deleting the root element?	02	1	2
1.9	For the input array [1, 2, 3, null, 4, null, null, 5, 6, null, 7] where index starts from 0, draw a binary tree.	02	2	2
1.10	An AVL tree node has a balance factor of -2. Identify the possible imbalance case(s) and the corresponding rotation(s) needed.	02	2	3
1.11	Analyze the function to search a key element in a Singly Linked List. Identify any errors or write the expected behavior. <pre>void searchElement(struct Node *head, int key){     while(head != NULL){         if(head-&gt;data == key){             printf("Found\n");             return;         }         head = head-&gt;next;     }     printf("Not Found\n"); }</pre>	02	1	4

### Part B

Question No	Question	M	CO	BT
2a	Implement C program to perform Push(), Pop(), and Display() operations on a stack including its main() function using array.	10	3	3
2b	Write an algorithm that uses stack operations to convert a infix expression into its prefix form. Explain with an example	06	3	3
3a	Develop a C program to implement insertion, deletion, and display operations on a circular queue using structures and pointers.	08	2	3
3b	Develop a C program to implement insertion, deletion, and display operations on Linear queue using structures and pointers.	08	2	2
<b>OR</b>				
4a	An online cart stores items (ID, price) in a Singly Linked List. Write C-style code snippets to perform the following operations: i) Insert products at beginning. ii) Calculate total cart value. iii) Display all items.	08	3	3
4b	Develop a c program to implement message queue using circular queue (with structures and pointers)	08	3	3



Write C-like code fragments for a Doubly Linked List (DLL) to accomplish the following tasks:

- 5a i) Use pointer manipulation to traverse the list and check whether the DLL contains a loop.  
 ii) Apply a two-pointer technique (moving at different speeds) to identify the existence of a cycle in the list, explaining each step of the process.  
 iii) Draw a neat diagram to demonstrate how the pointers move during loop detection and clearly state the condition that confirms the presence of a loop 08 1 4
- 5b Develop a C program to append one doubly linked list to another. Clearly explain how the prev and next pointers are adjusted during the merging process. 08 2 2

**OR**

- 6a An elevator scheduling system must continuously process floor requests in both upward and downward directions. Explain how a Circular Doubly Linked List (CDLL) can be used to manage elevator requests cyclically. Discuss how bidirectional traversal supports efficient handling of upward and downward movement. 08 2 4
- 6b Explain how polynomial operations can be carried out using a Doubly Linked List with the help of a C program. 08 4 3

Given the elements: 60,25,50,15,20,45,30,10

- 7a (i) Construct a max-heap using the top-down (insertion) method  
 (ii) Perform one iteration of Heap Sort  
 (iii) Explain heap restoration  
 Write the following C functions for a Binary Search tree 08 1 3
- 7b i) Recursive function to search for a key value in a Binary search tree  
 ii) To count all leaf nodes in a tree 08 3 3

**OR**

Given the elements: 45,20,35,10,25,15,30,5

- 8a (i) Construct a max-heap using the bottom-up approach.  
 (ii) Perform two iterations of Heap Sort by deleting the root element twice.  
 (iii) Show the heap after each heapify operation.  
 Write the following functions for a Binary Search tree 08 2 3
- 8b i) Function for postorder traversal.  
 ii) Recursive code to count non-leaf nodes in a Binary Search tree 08 3 2
- 9a Explain the AVL tree insertion algorithm. With an example. 08 2 2
- 9b A search engine tracks popular queries dynamically. Design a system using splay trees and justify its suitability. 08 3 5

**OR**

- 10a Explain why no single collision resolution strategy is optimal for all hashing applications. 10 3 4
- 10b Explain the Zig, Zig-Zig, and Zig-Zag operations in splay trees with suitable scenarios. 06 1 3