



# VIT

Vellore Institute of Technology  
(Decreed in the University under section 3 of UGC Act, 1956)  
CHENNAI

Reg. Number:

## Continuous Assessment Test (CAT) – I AUGUST 2025

|                            |                                                                                                                                                                                                                                  |              |                                                                                                                                                                                                                 |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Programme                  | : B.Tech                                                                                                                                                                                                                         | Semester     | : Fall Semester 2025-26                                                                                                                                                                                         |
| Course Code & Course Title | : BCSE202L & Data Structures and algorithms                                                                                                                                                                                      | Class Number | : CH2025260101955<br>CH2025260101953<br>CH2025260101957<br>CH2025260101959<br>CH2025260101960<br>CH2025260101376<br>CH2025260101489<br>CH2025260101706<br>CH2025260101700<br>CH2025260101703<br>CH2025260101418 |
| Faculty                    | : Dr.MANIMEGALAI<br>Dr.SENTHIL KUMAR A M<br>Dr.DOMINIC SAVIO M<br>Dr.ELAKIA E<br>Dr.ROLLA<br>SUBRAHMANYAM<br>Dr.KSHMA TRIVEDI<br>Dr.VIVEKANANDAN M<br>Dr.AFRUZA BEGUM<br>Dr.KARTHIKEYAN N<br>Dr.VALARMATHI K<br>Dr.KANTHIMATHI S | Slot         | : B2+TB2                                                                                                                                                                                                        |
| Duration                   | : 90 Minutes                                                                                                                                                                                                                     | Max. Mark    | : 50                                                                                                                                                                                                            |

### General Instructions:

- Write only your registration number on the question paper in the box provided and do not write other information
- If any assumptions are required, assume the same and mention those assumptions in the answer script
- Only non-programmable calculator without storage is permitted

### Answer all questions

| Q. No | Sub Sec. | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Marks | CO  | BT Level |
|-------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|----------|
| 1     |          | <p>You have a list of 7-digit numbers. To compare two numbers, use the Odd-Add-Even-Subtract rule:</p> <ul style="list-style-type: none"> <li>• Sum the digits in odd positions (1<sup>st</sup>, 3<sup>rd</sup>, 5<sup>th</sup>, 7<sup>th</sup>)</li> <li>• Sum the digits in even positions (2<sup>nd</sup>, 4<sup>th</sup>, 6<sup>th</sup>)</li> <li>• Calculate score = (sum of odd-position digits) - (sum of even-position digits).</li> </ul> <p>Design a pseudocode to sort the numbers by comparing their scores using Odd-Add-Even-Subtract rule.<br/>Example of Odd-Add-Even-Subtract rule:<br/><b>5382491</b></p> | 10    | CO3 | K3       |

|   |    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |     |     |    |
|---|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|----|
|   |    | <p>Odd positions: <math>5 + 8 + 4 + 1 = 18</math><br/> Even positions: <math>3 + 2 + 9 = 14</math><br/> Score = <math>18 - 14 = 4</math><br/> Input: 7408529 1357046 6043210 2345071 7123406 3456021 4561030<br/> Output: 2345071 3456021 1357046 4561030 7408529 6043210 7123406</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |     |     |    |
| 2 |    | <p>A cinema hall maintains an array <i>seats</i> with '<i>N</i>' size, where each element stores the <i>seats</i> status (0 for empty, 1 for booked). Customers can book a seat only if it is empty, and can cancel a booking only if it is already booked. Special Rule: After every 5 bookings, the system automatically shifts all booked seats to the front of the array (keeping their original order), followed by all empty seats, to optimize allocation for the next show. Formulate a pseudocode to handle booking, cancellation, and the 5-booking rearrangement rule using only arrays. Also find the time complexity of your pseudocode.</p>                                                                                                                                                                                                                              | 10  | CO2 | K2 |
| 3 | i  | <p>Solve the following recurrence relation using Masters theorem.<br/> <math>T(n) = 4 T(n/2) + n</math></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 4   | CO1 | K3 |
|   | ii | <p>Frame the recurrence relation and compute the time complexity of the following algorithm.</p> <pre>void test(int n)     if(n&gt;1)         for( i =0 ; i &lt; n; i++)             print n;         end for         test(n/2);         test(n/2);     endif</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 6   | CO1 | K3 |
| 4 |    | <p>A university maintains an array of student <i>CAT scores</i>, which may be unsorted initially. After each admission cycle, thousands of new scores are added, and the array is then sorted. The exam office frequently needs to search for specific scores to determine if students qualify for the next round. To improve efficiency, design a pseudocode using <i>Divide-and-conquer approach</i> that first sorts the array (if unsorted) and then searches for a given CAT score. Also your pseudocode should return the index where the score is found, along with the total number of comparisons made during the search.</p> <p><b>For example</b>, given the CAT scores [55, 68, 72, 81, 90, 94, 100, 115, 130, 145, 160, 175, 190], if a student wants to check for the score 94, the output should be: <b>"The score 94 is found at index 5 after 4 comparisons."</b></p> | 10  | CO3 | K3 |
| 5 |    | <p>In a computer operating system, a resource manager uses a formula to calculate the CPU load balancing factor for different processes to ensure efficient processor utilization.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 7+3 | CO2 | K2 |

|                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |  |  |  |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|--|
|                         | <p>To optimize scheduling, the operating system uses the following infix expression:</p> $((A + B * C) - (D * E + F)) / (G + H - I)$ <p>However, since the operating system uses a stack-based Reverse Polish Notation (RPN) evaluator, this expression must first be converted to RPN. Then, evaluate the RPN using stack operations with the following values:</p> <p>A=2, B=1, C=2, D=1, E=1, F=1, G=3, H=1, I=2</p> <p>Show the illustration of each operation in the stack during both RPN formation and evaluation.</p> |  |  |  |
| *****All the best ***** |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |  |  |  |