REG.NO.:

VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering
CONTINUOUS ASSESSMENT TEST - II
WINTER SEMESTER 2025-2026

SLOT: G2 + TG2

Programme Name & Branch	: B. Tech Computer Science and Engineering
Course Code and Course Name	: BCSE305L, Embedded Systems
Faculty Name(s)	: Common to all
Class Number(s)	: Common to all
Date of Examination	: 23/03/2026
Exam Duration	: 90 minutes

Maximum Marks: 50

- **General instruction(s):**
- Answer All Questions
- M- Max mark; CO – Course Outcome; BL – Blooms Taxonomy Level (1 – Remember, 2 – Understand, 3 – Apply, 4 – Analyse, 5 – Evaluate, 6 – Create)
- Course Outcomes:

3. To examine the working principle and interface of typical embedded system components, create programme models; apply various optimization approaches including simulation environment and demonstration using debugging tools.
4. To evaluate the working principle of serial communication protocols and their proper use, as well as to analyze the benefits and drawbacks of real-time scheduling algorithms and to recommend acceptable solutions for specific challenges.

Q.No	Question	M	CO	BL

1.	Given the following sequence of expressions:
	a = q - r;
	b = a + t;
	a = r + s;
	c = t - u;
	Construct the Data Flow Graph (DFG) showing variable dependencies and redefinitions across the sequence. Perform optimal register allocation with the aid of lifetime graph and graph coloring problem to determine the minimum number of registers required.
		10	3	4

2.	Design a Control/Data Flow Graph (CDFG) and a Finite State Machine (FSM) for a water pumping system where water is transferred from a sump to a tank of height 18 cm. The pump should operate until the tank reaches a water level of 15 cm, at which point it stops. Also, identify the essential hardware components required to implement this prototype.	10	3	3
3.	Apply your findings on the given code snippet to get the optimized code with relevant validations *#include <stdio.h>* *#define SIZE 100* *int main() {* *int sensor[SIZE];* *int i;* *int sum = 0;* *int square_sum = 0;* *// Initialize sensor values*	10	3	3

```c
    for (i = 0; i< SIZE; i++) {
        sensor[i] = i * 2;   // simulate sensor data
    }

    // Compute sum
    for (i = 0; i< SIZE; i++) {
        sum += sensor[i];
    }

    // Compute square sum
    for (i = 0; i< SIZE; i++) {
        square_sum += sensor[i] * sensor[i];
    }

    printf("Sum = %d, Square Sum = %d\n", sum, square_sum);
    return 0;
}
```

3. For the set of periodic processes given below:

Process	Execution Time	Period
P1	1	3
P2	1	4
P3	2	6

a) Schedule the processes using the Rate Monotonic Scheduling (RMS) algorithm over a time interval equal to the least common multiple (LCM) of their periods. Assign priorities according to the RMS policy, where the process with the shorter period is given higher priority. Also compute Utilization and check bound test.

b) Schedule the processes using the Least Slack Time (LST) algorithm. At each time instant, dynamically assign priority to the process with the least slack time. Construct the schedule until the slack time of any process becomes zero

10	4	4

5. In a POSIX-based dashboard system where a speed sensor process and a fuel sensor process send their status updates to the parent dashboard process, explain how POSIX features such as fork(), pipe(), read(), write(), wait(), copy-on-write memory, scheduling, and process queues are used in practice. How do these features coordinate the parent and child processes to ensure that speed and fuel data are safely transmitted, synchronized, and displayed on the dashboard?

10	4	5

****