Adv C Module

1.Subjective

2.1 Basic Refreshers

1. Explain real-time examples for If, Else if, and Switch cases.
2. What is the difference between embedded c and c
3. Explain the bitwise operator.
4. How should we access elements in an array?
5. Write C code to convert 2890 into binary and count How many bit is set.
6. Explain the size of all the data types.

2.2 1D Pointers and Functions

1. Explain the difference between pass-by-value and pass-by-reference with an example program.
2. Explain the use of pointer.

2.3 String

1. Explain the string library function in detail.
2. Explain the strcat, strcmp, strlen, strstr, strchar and strtok in detail.

2.5 Storage classes and memory segments

1. Explain static variable and linkage of that.
2. What is meant by overflow?
3. Explain the volatile keyword.
4. What is a dangling pointer, and how can it be overcome?
5. Explain the different types of memory segments.
6. Explain the volatile and const keywords.
7. Explain Difference in Const int* a, int* const a

2.6 2D Pointers and DMA

1. Explain the dangling pointer with an example program. How we overcome that?
2. What is the use of a function pointer?

3. Explain the DMA in detail.
4. Explain the DMA with an example.
5. Explain the use of a double pointer.
6. Explain the dangling pointer.

2.7 Preprocessing
1. What is the use of #include?

2.8 UDT
1. Explain the structure and union in detail.
2. Explain the use of structure.

2.9 Miscellaneous
1. Explain the Volatile keyword with an example program.
2. List some applications of volatile keywords.

2.10 FILE I/O
1. Explain the fopen, fread, fwrite and fseek in detail.

2. Programming

1. WAP to count the no.of set bits, If it is even reverse the bits.
2. Wap to count no of set bits if it's even swap those bits.
3. Write a single line code to take 3 values from the user. All three should belong to different data types.
4. Write a program to check whether the number entered by the user is even or odd.
5. WAP to find the greatest of three numbers.
6. Write a program to print the following star pattern:
7. Write a program to add the digits of a given number
8. Write a program to check if a number is prime or not
9. Write a program to print prime numbers between m and n
10. Write a program to print n prime numbers
11. Write a program to find the sum of all the elements in the list

12. Write a program to print odd numbers in a given range, m to n
13. Write a programme to reverse a string and check if the string is a palidrome or not.
14. Write code where I pass four integer values and return the sum and difference of four numbers
15. WAP to reverse the string without using a loop.
16. WAP to find odd or even without using '%'.
17. WAP to add two numbers without using '+'.

MC Module

1. Basic electronics
1. Explain the CLCD?
2. Explain the architecture of PIC.
3. What you learned from your PIC.

2. Basics
1. Explain the difference between C and Embedded C?
2. How do you calculate the speed of the rotating disc using a microcontroller?
3. What is meant by bidirectional ports?
4. What is the difference between ROM and RAM
5. WAP will generate a 1-sec delay.
6. Difference between microcontrollers and microprocessor

3. Interrupts
1. Explain the difference between Counter and Timer.
2. What is meant by interrupt?
3. Explain the use of ISR.

4. Projects
1. What are the difficulties you faced in the car block box?
2. What are the protocols used inthe Car black box project?

5. ADC
1. Explain ADC and how it is working. How many bit ADCs did you use?

6. Embedded Systems

1. Explain the components of embedded systems in details.

7. Protocols

1. How can you know whether data is successfully transmitted in the UART protocol?
2. Why do we need to send 2 stop bits in the UART protocol?
3. Explain SPI protocol.
4. What is meant by the baud rate?
5. Give me the real-time applications of protocols.
6. What is the minimum and maximum number of bits that can be transmitted in UART?
7. Explain the UART protocol with a frame diagram.
8. How can you know whether the data is transmitted or not?
9. Why did we need to transfer 2 stop bits?
10. How many bit ADC did you use and how to convert analog to digital
11. Configure the TM4C123GH6PM to interface with an RTC module over I2C.
12. The date and time should be set or adjusted via a switch press. Periodically, the current date and time should be read from the RTC, and the values should be sent over UART upon user request.
13. Explain UART coding.
14. Difference between the UART and USART.
15. LCD display, which protocol was used?
16. Protocols in Detail [Data frames]

8. PWM

1. How to control the brightness of LED?

9. programming

1. Develop firmware to achieve the following functionality:

1. **Switch 1 Functionality**: Each press of Switch 1 cycles the RGB LED through different colors. After the last color, it rolls back to the first.
2. **Switch 2 Functionality**: Pressing Switch 2 triggers an interrupt that starts a timer. The timer reads ADC data (e.g., from a potentiometer) every 5 seconds and sends the ADC value to the terminal via UART. Pressing Switch 2 again restarts this process.

CPP Module

1. Explain the qualifiers and Modifiers.
2. (in C++) you have 2 arrays that will be in a sorted order in that some common element may be there (Find the sum of (max to min) of that array). You can start from one array to another
3. Explain the use CLASS
4. Explain function overloading.
5. Difference between the structure and class.
6. How will we call the cpp function in the qml file
7. Methods to expose cpp file to qml
8. They asked one class and how will declare this calss using unique pointer
9. They gave 2 cplu plus code and wrote the output for that
10. Methods for QAbstraclistmodel
11. How will start the thread on cpp
12. Which property will you use to insert one rectangle next to another rectangle
13. and connect object syntax
14. How will you write the syntax for declaring the var Q property as a Boolean variable?
15. Explain the pillars of OOPs.

16.   Explain the inheritance in detail.

Ds Module

1. Basics

1.   What is meant by the resolution of a register?

2. Linked lists

1.   WAP swaps the two nodes in the single linked list.
2.   WAP swaps the two nodes without a third member in the single linked list.
3.   Explain the use of single linked list.
4.   Explain the difference between the single linked list and double linked list.

3. Stack

1.   Explain the use of stack in detail.
2.   Explain the stack application.

4. Searching and Sorting Techniques

1.   Explain the sorting and searching algorithms.

5. Queue

1.   Explain the use of queue

6. Hashing

1.   What is meant by hashtable, what was used, and where can we use it?

7. Trees

1.   Where we will apply tree concepts in real life.
2.   Explain the binary search tree.

LI Module

1. Basics

1. How to change permission in Linux
2. Explain the 755 permission
3. Explain the Linux kernel.
4. Explain the difference between user space and kernel space.
5. Explain layers of the OSI model.
6. Explain layers?
7. What is os.
8. Tell the command to remove the file and directory.
9. Explain the use of kernel space.
10. Explain BIOS

2. System call

1. Explain the use of system calls.

3. Networking

1. Explain the difference between TCP and UDP.

4. Process

1. How to check the current running process in Linux.
2. Explain the use of the fork system call.

5. IPC

1. Explain the use of the IPC mechanism in detail.

6. Signal

1. Explain the signal handler.
2. Explain the signal handler and sigaction.
3. Explain the SIGALRM signal.
4. Explain the SIGCHLD signal.

7. Socket

1. Explain the TCP socket and UDP socket.

8. Threads

1. Explain the difference between the Process and Thread.
2. Explain the race condition in detail.

3. Explain the critical section.
4. How to do synchronisation in thread.

9. Process and Memory Management
1. Explain the process and memory management.

Qt Module

1. Write C++ code in QT. Get user input and output should be stored in file
2. Experience debug methods
3. How does qdebug impact
4. How does Qwarning impact
5. Explain the device driver.
6. Which property is used to insert one rectangle next to another rectangle

General Questions

1. Implement the firmware to change the RGB LED color each time Switch 1 is pressed.