Adv C Module

1. Subjective

2.1 Basic Refreshers

1. Explain binary conversion with examples.
2. Who collects the return value from the main() function?
3. What is the size of int?
4. What is the size of char?
5. What is the size of the float?

2.2 1D Pointers and Functions

1. Explain the type of pointer.
2. What is the difference between pass by value and pass by reference?

2.3 Storage classes and memory segments

1. Explain the storage class in detail.
2. What is the difference between static and extern?

2.4 2D Pointers and DMA

1. Explain the memory segments.
2. Explain memory segments in a program (code, data, stack, heap).
3. Explain dynamic memory allocation.
4. Explain function pointers in C and their usage with examples.
5. What is the difference between malloc and calloc?

2.5 Preprocessing

1. Explain the compilation stages.

2.6 UDT

1. What is the difference between a structure and a union?
2. How do you set a bit using a macro?
3. How do you clear a bit using a macro?
4. What is meant by UDT?

2.7 Miscellaneous

1. What is the volatile keyword in C?

2.8 Project

1. Explain the MP3 tag reader project.
2. Explain the steganography project.
3. Explain the address book project.
4. Where is steganography used, and where is cryptography used?

Programming

1. Write a program to print n prime numbers (it should work for any big number).
2. WAP to find the largest among three numbers.
3. Write a program to concatenate two strings.
4. Write a program to generate the fibonacci series.
5. Write a program to rearrange the bits of an 8-bit binary number such that all 0's are moved to the left and all 1's to the right.
6. WAP to check whether the system is little endian or big endian.
7. WAP to reverse the string.
8. Write a C program to get the information of 5 students using a structure and dynamic memory allocation (malloc).
9. WAP to count 1s and 0s in an integer.
10. Write a program to read integer data from a file, square each integer, and write the squared values to another file.
11. WAP to check if given is a palindrome or not.
12. WAP to check if a given number is a power of 2 or not.
13. Write a program to check whether a given number is a power of 4.
14. Write a program to print array elements using pointers.
15. Write a program to copy the contents of one file to another.
16. If you are given a string "0x1234" write a program to print its hexadecimal value.
17. WAP to implement the strcmp().
18. WAP to implement the strcat(). Explain the usage of the strcat() function.
19. Given an 8-bit binary number (e.g., 10101010), write a program to perform the following operations:

- If the number is even, toggle (invert) all the bits.
- If the number is odd, reverse the order of all bits.

How will you implement this?

MC Module

1. Basic electronics

1. What are the types of transistors?
2. What is a Zener diode?

2. Basics

1. What is the interfacing component used between a potentiometer and a microcontroller?
2. Draw the microcontroller architecture.
3. How do you calculate the speed of the rotating disc using a microcontroller?
4. What is the difference between C and embedded C?
5. Explain the difference between MC and MP.
6. Explain the difference between RAM and ROM.
7. What is the difference between edge-triggered and level-triggered interrupts?
8. How does a microcontroller handle multiple interrupts occurring at the same time?

3. Interrupts

1. What is an interrupt?
2. Explain the ISR and IVT.
3. Is the interrupt service routine (ISR) configured as edge-triggered or level-triggered, and at what condition (rising/falling edge or level) does the interrupt occur?
4. If a timer interrupt is configured to trigger every 5 ms, but inside the ISR the task/code takes 10 ms to execute, how frequently will the ISR actually be called, and why?

4. Projects

1. Explain Car black box with a block diagram.
2. Explain the Pick-to-Light project.
3. Explain the CAN based Automotive Dashboard project.

5. ADC

1. Also, explain different modulation types with a special focus on QPSK. What makes QPSK special compared to other schemes?
2. What is the difference between 8-bit and 10-bit ADC?

6. Embedded Systems

1. What is an embedded system? Explain its characteristics and give two real-time examples.
2. What is the role of a watchdog timer in embedded systems?
3. What are the different types of memory used in embedded systems?

7. Protocols

1. How do you communicate with an EEPROM using the I2C protocol?
2. How is CAN communication used in a Pick-to-Light system?
3. How does arbitration work in CAN communication?
4. What is the maximum data transfer distance for UART communication, and why is it limited?
5. What is the maximum baud rate supported by UART communication? What happens if we exceed this limit?
6. Explain the UART protocol with a frame format. Write a program to identify the header and data length from a given frame and extract the data into another buffer based on the identified length.
7. Explain the I2C protocol with a frame format. Explain the data attribution.
8. Explain the CAN protocol with a frame format. Write a program to identify the header and data length from a given frame and extract the data into another buffer based on the identified length.
9. Explain the SPI protocol with a frame format.
10. What is the difference between simplex and duplex communication?
11. Explain UART and USART.

8. PWM

1. Define duty cycle with suitable examples.

9. Programming

1. You are given three I2C addresses: 0x01, 0x02, and 0x03. Write a program to read data from these addresses using the I2C protocol.
   When an interrupt occurs:
   - If the interrupt source is 0x01, toggle between play and pause of a song.
   - If the interrupt source is 0x02, clear the interrupt.
   - If the interrupt source is 0x03, turn on the backlight for 3 seconds. If the button is pressed again within this time, turn off the backlight for 3 seconds.

   Explain how you would implement this logic.
2. In a project, two devices are connected via UART. You are provided with driver functions like getbyte() and write(). Using this information, write two functions—TX() and RX().
   In TX():
   - You must create a data packet in the following format:
   - [Header (2 bytes)] [Event (1 byte)] [Size (2 bytes)] [Data (size bytes)] [Footer (2 bytes)]
   - The header and footer are fixed hexadecimal values.
   - The event is a 1-byte identifier.
   - The size represents the number of data bytes after removing any NULL characters from the data.
   - After creating the packet, send it using the write() function.

   In RX():
   - You must read the incoming packet using getbyte().
   - Extract the event and data from the received packet.
   - Pass the extracted event and data to the dosomething() function.

   Write the code for both TX() and RX() based on the above requirements.

CPP Module

1. What is the difference between a virtual and a pure virtual function?
2. Explain the OOP concepts with an example.
3. WAP to create a linked list using C++.
4. What should we do to force a derived class to override a base class function? Explain.
5. Explain the types of constructors.
6. What is the difference between C and CPP?
7. What is the difference between a shallow copy and a deep copy?
8. What is the difference between a structure and a class?
9. What is a copy constructor? What are its types?
10. What are virtual functions?
11. Explain the Polymorphism in detail.
12. What is polymorphism?
13. What is a V-Table (Virtual Table)?
14. What is the relationship between a V-Table and inheritance?
15. Explain smart pointers.
16. Explain operator overloading of new and delete operators.

Ds Module

1. Basics

1. What is the difference between LIFO and FIFO? What are their applications?

2. Linked lists

1. What is a linked list? How does it differ from an array? Why we use it and its advantages.
2. What are some real-time applications of linked lists?
3. Write a program to add, delete, and search an element in a singly linked list.
4. Write a logic to find the unsorted node in a sorted linked list.

5. Write a function to remove a node from a linked list when only the node pointer is given.
6. Write a function to insert a node at a given index in a linked list.
7. How are linked lists used in web browsers (e.g., navigation history)?
8. How are linked lists used in text editors (e.g., undo/redo operations)?

3. Stack

1. Explain the stack in detail.

4. Searching and Sorting Techniques

1. What is the difference between linear search and binary search? Explain with examples.
2. Explain the working of the bubble sort algorithm. What is its time complexity?
3. What is quicksort? Explain its algorithm and best, average, and worst-case time complexity.
4. What is the difference between stable and unstable sorting algorithms? Give examples.
5. Which searching or sorting technique is preferred when data is nearly sorted? Why?

5. Queue

1. What is a queue? Explain its operations with examples.
2. What is the difference between a queue and a stack?
3. Explain the concept of a circular queue. How does it prevent overflow?
4. What is a priority queue? Where is it used?

6. Hashing

1. What is a hash table? Explain with an example.
2. What are some real-time applications of hashing?

7. Projects

1. Explain the inverted search project.
2. Explain the APC project.

8. Trees

1. What is a binary tree? How is it different from a binary search tree (BST)?
2. Explain inorder, preorder, and postorder traversals of a binary tree.
3. What is a balanced binary tree? Give examples.
4. What is a heap tree? How is it used in heap sort?

LI Module

1. Basics

1. What are an Operating System (OS) and a Kernel?
2. What do you know about kernel bootloading?

2. System call

1. What is a system call? How is it different from a normal function call?
2. Give examples of common system calls in operating systems.
   (e.g., read(), write(), open(), close(), fork(), exec(), exit())
3. What is the role of a system call in the context of user space and kernel space?
4. Explain the difference between fork() and exec() system calls in UNIX/Linux.

3. Networking

1. Questions on the SMB (Server Message Block) protocol—its purpose and where it is used.
2. Questions on HTTPS (Hypertext Transfer Protocol Secure)—its importance and how it ensures secure communication.
3. They also asked about the protocols used in macOS for networking and communication.

4. Process

1. What is a process in an operating system? How is it different from a program?
2. What are the different states of a process? Explain each.

5. IPC

1. Explain the IPC mechanisms and their applications.
2. What is the difference between unnamed pipes and named pipes (FIFOs)?
3. What are the advantages and disadvantages of shared memory IPC?

6. Signal

1. What is a signal in an operating system?
2. How does a signal differ from an interrupt?
3. Give examples of commonly used UNIX/Linux signals.
   (e.g., SIGINT, SIGKILL, SIGTERM, SIGSEGV)

7. Socket

1. What is a socket? How is it used in networking?
2. What is the difference between TCP and UDP sockets?
3. Explain the server-client model using sockets.

8. Threads

1. Explain the threads in detail.
2. How do you create a thread in C or C++?
3. Explain the multithreading in detail and synchronization methods.
4. What is the difference between mutexes and semaphores?
5. What is the difference between a thread and a process? Explain with examples.
6. What is deadlock?
7. How can we avoid the race conditions?

9. Projects

1. Explain the minishell project.
2. Explain the TFTP project.

10. Process and Memory Management

1. What is the difference between physical memory and virtual memory?
2. Explain paging and segmentation in memory management.
3. What is a page fault, and how is it handled by the OS?

General Questions
Aptitude

1. Explain these commands, like top, more, ps, vi, and free, and their usage.
2. You have a cake. You are allowed to make only 3 cuts. How can you cut the cake into 8 equal pieces using just these 3 cuts?
3. What protocols or antennas are used in TVs and mobile towers?
4. What are the types of antennas? Explain each.
5. Explain 4G, VoLTE, and LTE.
6. Explain filters: Low-pass, High-pass, Band-pass, and Band-stop.
7. Explain frequency, wavelength, and their relationship.
8. Explain about U-Boot.
9. What is a kernel image?
10. What is yocto project.
11. You are given some prewritten functions to check the availability of Wi-Fi and cellular data. Write a program that activates Wi-Fi if available; if Wi-Fi is not available, activate cellular data. If neither is available, display an error message. The status (Wi-Fi ON, Cellular ON, Error) must be displayed on an LCD.
12. Explain back EMF and why it is considered an opposing voltage in motors.
13. Explain why transformer ratings are given in kVA instead of kW.
14. Explain whether DC can be applied to transformers and justify your answer.
15. Explain the significance of using a 4–20 mA current loop in industrial applications.
16. Explain what an inverter is and its basic working principle.
17. Define an operational amplifier (Op-Amp) and explain its basic operation.

18. Share your learning experience at Emertxe.
19. Why did you choose the Electronics branch in engineering?
20. Explain basic DC and AC circuits.
21. Draw and explain the BJT (Bipolar Junction Transistor) diagram.
22. Explain XOR and NAND gates, and why is the NAND gate called a universal gate?
23. Explain the SDLC process.
24. What is the difference between software development and software testing?
25. What are the different types of software testing?
26. Explain the difference between NPN and PNP transistors.
27. What type of interrupt is used for interfacing with the HT9170 IC?
28. To which port (or pin) is the HT9170 connected in the microcontroller circuit?
29. Assignment:

The following IC is to be interfaced with a microcontroller.

- HT9170

1. Explain the use of the IC mentioned above.
2. Explain how the IC gives the output.
3. Make a block diagram showing the interfacing with any 8051 / Any AVR Microcontroller/ Any STM controller.
4. Write a code in the Embedded C Language to achieve the following:
    a. The microcontroller is clocked using an external crystal of 16 MHz. The clock setup needs to be done manually and not by using any library function.
    b. DTMF Reception Logic: - Assume a valid DTMF can be received at any point in time. - Read the valid DTMF data. - Store the first 10 digits in an array. - Use interrupts to achieve this logic