Adv C Module

1. Subjective
2.1 Basic Refreshers

1. Explain the
2. What is the final value of n?

   ```
   int n = 64 ;
   n=n++;
   n=n<<4;
   n=n|12;
   n=n>>2;
   n=++n;
   ```
3. Why do we use the bitwise operator?
4. What is the difference between logical && and bitwise &?
5. Explain the different types of data types.
6. Working of assignment operators
7. Which system is better?
   1. 2x32
   2. 1x64
   3. 4x16
   4. All Same
8. Which one has a high precedence (* , / , %)?
9. int ptr = 100;
   ptr++;
   printf("%d",ptr);
10. int *ptr = 100;
11. ptr++;
12. printf("%d",*ptr);
13. int *ptr = 100;
    ptr++;
    printf("%d",&ptr);
14. Int i=1;
    While(i++)

```
{
    printf("hello");
}
```
He asked What will happen with is code

15. What is the use of logical operators?
16. What are bitwise operators?
17. What are data types, and why do we need them?
18. What characteristics are defined by data types?
19. What is the difference between int and char?
20. What is the difference between logical AND and bitwise AND?
21. What are looping statements?
22. What is the difference between i++ and ++i?
23. Can we define a function in .h file?
24. What is the use of the (%) module operator?

2.2 1D Pointers and Functions

1. What is the size of a void pointer and an integer pointer?
2. Explain the pass-by-value and pass-by reference.
3. What is the use of a pointer?

2.3 String

1. Explain the string function in detail.

2.5 Storage classes and memory segments

1. What is the use of the global variable and extern keyword?
2. What is the diff between a global variable and a static global variable?
3. Why do we use the volatile keyword and const keyword?
4. Briefly explain about storage classes
5. What is a segmentation fault?
6. How to access a variable declared in another file.

2.6 2D Pointers and DMA

1. What is the diff between a pointer to an array and an array of a pointer?

2. Write the syntax for collecting the address of the structure in another function.
3. Write a syntax for function pointer?
4. What are void pointers, wild pointers, and dangling pointers?
5. What is the NULL pointer
6. Why we need to use DMA in detail. What is the use of it?
7. Write the syntax for malloc(), calloc(), and realloc().
8. What is a volatile keyword?
9. What is a const keyword?
10. Can we use volatile and const keywords at the same time?
11. Explain the calloc and malloc functions using a void pointer.
12. Explain the use of virtual functions.
13. Explain about function overriding.

2.7 Preprocessing

1. What is macro and WAP to swap two numbers using macro?

2.8 UDT

1. Explain the difference between the structure and bitfield structure.
2. Explain the use of typedef.
3. Write a program to return a structure.
4. What is enum? how can we make an enum a pointer type?
5. Write a structure. and create a structure array. pass the structure into a function
6. Explain the difference between the structure and Union.
7. what is the size of this structure
```
struct node
{
    Int a;
    struct node *p;
}
```

2.9 Miscellaneous

1. Why do we use volatile and const keywords?
2. Can we use const and volatile together?
3. What is volatile?

2. Programming

1. WAP is used to convert the decimal number to binary.
2. WAP to convert the binary to decimal.
3. Pattern printing (if N = 4)

   ```
   1
   2 5
   3 6 8
   4 7 9 10
   ```
4. WAP to find the factorial using function.
5. WAP to convert a number to binary and count the number of 0s and 1s in it.
6. WAP to toggle the bits.
7. WAP to reverse the number.
8. WAP to reverse the string.
9. Write the string copy function.
10. WAP to convert lower case letters to upper case letters.
11. WAP to insert the new value at a specific position In the array
12. Write a program to read an array and find the largest and second-largest number.
13. WAP to calculate the factorial of a number using a function.
14. WAP to read a decimal and print it in hexa and binary.
15. WAP to print binary format for the given number
16. swap nibbles in macro
17. Static variable program
18. WAP to reverse a number.
19. WAP to read a number and convert the 0s to 1s and 1s to 0s.

20. WAP to print the array value using a pointer.
21. WAP to check if the given number is a power of 2 or not.

MC Module

1. Basic electronics

1. How can we access the microcontroller using C?
2. How do you access the peripherals using C code?

2. Basics

1. What is RTC? Explain in detail.
2. How to read and write the content in CLCD.
3. What is the protocol used in PIC18F4580.

3. Interrupts

1. What is the difference between an interrupt and a polling mechanism?
2. They asked about interrupts in microcontrollers and the ISR function. In the ISR function, which kind of lock would you prefer to block the content in the ISR function?
3. Explain the watchdog timer in detail.
4. Explain the clock synchronization and stretching.

4. Projects

1. Explain the car black box project.

5. ADC

1. Explain how the ADC is working and explain how to use.

6. Protocols

1. What is the difference between synchronous and asynchronous communication protocols?
2. Which is faster, UART or SPI, and why?
3.   How many pins are present in I2C?
4. Why do we use I2C?
5. What is the concept of multi-master and multi-slave?
6. What is a half-duplex?

7. Explain about protocols
8. Explain the UART protocol.
9. Explain the I2C protocol.
10. Explain the SPI protocol.
11. Explain the difference between SPI and I2C protocol.
12. Explain the difference between UART and I2C protocol

7. PWM

1. Explain where we will use PWM.

CPP Module

1. Explain the inheritance Concept in detail.
2. Implementation of linked list using C++.
3. Implementation of annagram.
4. Explain the OOPs concept.
5. Initiate and start threading using C++.
6. Explain about Polymorphism in detail.
7. Which is least type casting is preferred in C++.
8. Explain the use of malloc and new.
9. Explain the constructor in detail.

Ds Module

1. Basics

1. Find time complexity for nlogn+logn+1.

2. Linked lists

1. Find Loop is present in the list or not.
2. Create the loop in the single linked list.
3. Find the middle value in the linked list.
4. Explain Double linked list.
5. Explain the difference between an array and linked list
6. WAP to find the mid node in linked list.
7. How many arguments are required for the circular linked list?

8. How to find a loop in detail.
9. WAP to print the last node data in SLL.
10. WAP to insert at first in SLL.

3. Stack

1. Explain the stack in detail.

4. Searching and Sorting Techniques

1. Explain the different types of searching is there.
2. Explain the different types of sorting.

5. Queue

1. Explain the queue in detail

6. Hashing

1. Explain hashing in detail.

7. Trees

1. Explain about tree concept.

LI Module

1. Basics

1. Explain the kernel space.
2. Explain the device drivers and subsystem.
3. What is BIOS?
4. How to change permission in Linux.
5. Explain 755 permission.
6. How to check the current running process in Linux.
7. What is the chmod command in Linux?
8. What is the ps command in Linux?
9. What is the difference between user and kernel spaces?

2. System call

1. Explain the about system call.

3. Networking

1. Explain about networking protocol in detail.
2. Explain the difference between IPv4 and IPv6.

4. Process

1. What is process, and how to know which process is running?
2. Explain the use of the fork system call.
3. Explain the use of exec system calls with variations.

5. IPC

1. Explain the IPC mechanism in detail.
2. Explain the different types of IPC.
3. Explain PIPE.
4. Explain FIFO.
5. Explain SHM.

6. Signal

1. Difference between the signal() and sigaction().
2. Explain the sa_flag?

7. Socket

1. Explain UDP and TCP/IP socket in detail

8. Threads

1. Explain the use of thread in detail.
2. Explain the race condition in detail.
3. Difference between semaphore and mutex.
4. Difference between thread and process.
5. Difference between mutex and binary semaphore
6. Explain Threads.
7. Explain how a thread performs multitasking at a time.
8. what is meant by threads how the thread will differ from the process.
9. What is multithreading?
10. What will happen when a thread crashes and when a process crashes?
11. Can we use semaphore in place of mutex?

9. Process and Memory Management

1. Explain process management and memory management.

Qt Module

1. syntax for connect function what is signal and slot

General Questions

1. Why do we use an Embedded system?
2. Why do we use embedded C?
3. Which languages can we use for embedded systems other than C?
4. What is a compiler error?
5. If you have a round cake and you are to didtribute it among 3 kids equally. But you need to ensure that no kids get more than 1/4th of the cake at a time. This cake can be used by the kids for 2 days. How much cake will each child get each day before the cake is finished? Please explain your work for the answer.
6. If the time on the clock is 3:15 PM, what will be the angle between the hour hand and the minute hand?
7. Assume you have a speaker system with 2 speakers and one subwoofer connected to your computer via Aux cable. There is no audio coming from your speakers. What will be your step-by-step process to check where the issue lies?