Adv C Module

1. Subjective

2.1 Basic Refreshers

1. Difference between C and Embedded C
2. How do you compile a C program on a Linux system?
3. What is GCC?
4. How will you add two numbers without using +.
5. What is the size of each data type?

2.2 1D Pointers and Functions

1. What is call by reference and call by value?
2. What is the purpose of the function?
3. What are the advantages of the function and pointer?
4. Can we perform the pass by reference without pointers?
5. How to access the variable using a pointer.

2.3 String

1. What is the return type of strcmp and strstr?
2. Implement the strstr function.
3. What is the difference between getc and getchar?

2.5 Storage classes and memory segments

1. Write the syntax for a const pointer.
2. Explain the storage classes in detail.
3. Explain the extern keyword in detail.
4. Explain the segments in C.
5. Explain the segmentation fault in pointers.
6. Explain the storage classes in detail
7. Explain the static keyword in detail.

2.6 2D Pointers and DMA

1. Explain the smart pointer in detail.
2. Explain the vector-related functions
3. Explain the dynamic memory allocation in detail.
4. Explain the memory allocation in malloc and calloc
5. Explain the dangling pointer

6. What is a Memory leak? Explain with an example.
7. How will you free the memory without using free()?
8. Explain the function pointers with examples. Explain the dynamic memory allocation. Explain DMA with examples.
9. What is a function pointer? Write the syntax of a function pointer.

2.7 Preprocessing

1. Explain the compilation stages with relevant commands.
2. Explain about macro.

2.8 UDT

1. Declare the structure and the size of this structure.
2. What is the difference between a structure and a union? Explain with a padding.
3. What will happen while declaring the structure if I assign a value?
4. Explain about bitfields in detail.

2.9 Miscellaneous

1. Explain the volatile keyword.
2. Explain the difference between const and volatile.

2.10 FILE I/O

1. Is the fopen system call OS independent?

2. Programming

1. WAP to print the prime numbers up to N.
2. WAP to print the fibonacci series upto N.
3. Write a macro to clear the nth bit of a number.
4. WAP to reverse the array elements.
5. WAP to find the target element in the sorted array. It returned the index.
6. WAP to print the name.
7. WAP to check whether the number is prime or not
8. WAP to check whether the given number is even or not.
9. Write a code to encode a character.

MC Module

1. Basic electronics

1. What code is executed in the microcontroller before the execution of the main function?
2. Explain the different software methods to reset an MC.
3. The microcontroller stores all the memory in RAM itself.
4. Why do we have registers?
5. List the differences between a microcontroller & microprocessor.
6. How pull-up and pull-down work.
7. Explain the EEPROM.
8. How many pins are there in pic18f4580?
9. What is the use of the Watchdog timer?
10. What are the transistor, filter, and diode?

2. Basics

1. Difference between a microcontroller and a microprocessor
2. Why did you choose the PIC microcontroller specifically?
3. What is a voltage conversion device? What is a MOSFET?
4. How can it be switched, and what are its terminals? Is the gate terminal similar to a regular logic gate?

3. Interrupts

1. Explain the interruption in detail.
2. How much is the clock frequency in 1 second?
3. List the differences between polling and interrupting. List the advantages and disadvantages of using interrupts instead of polling.
4. Explain how the timer is working. How to calculate 1 second using an 8-bit timer at 20MHz frequency.
5. What is going to be the count of the timer for generating a delay of 10 milliseconds in a controller running at a speed of 4 MHz?
6. Explain about timers and their types.

4. Projects

1. Explain the car black box project in detail
2. Explain the pick-to-light project in detail.
3. Explain the microwave oven project in detail. Ask for the code to start the microwave, then display "Microwave Started" on the CLCD and prompt the user to set the time. Then, write the program to set the time.

5.ADC

1. Explain the ADC in detail.
2. How to convert analog to digital signals?

6. Embedded Systems

1. Explain about an embedded system.

7. Protocols

1. What is the CAN protocol?
2. List out the hardware which are used in the CAN automotive project.
3. Explain the UART protocol with a frame format. Write a simple code.
4. Explain the SPI protocol with a frame format.
5. Explain the I2C protocol with a frame format.
6. Explain what is meant by the baud rate and where it is used.
7. Why should we communicate at a very high baud rate?
8. Describe 2 to 3 debugging steps if the UART protocols are not working.
9. What is the difference between UART and USART?
10. What are the disadvantages of UART communication?
11. What is the difference between SPI and I^{2}C protocols?

8.PWM

1. Explain how the PWM is working.

CPP Module

1. How will you create a user-defined template in C++? Give an example.
2. Explain the Pillars of OOPS.

3. What is STL in C++? Why do we use it?
4. What are the main components of STL?
5. Name a few commonly used STL containers.
6. What is the difference between a vector, a list, and a deque?
7. How does a map differ from an unordered_map?
8. What is the difference between a set and a multiset?
9. What is the difference between a struct and a class in C++?
10. What is the difference between new and malloc?
11. What are access specifiers in C++? Name them.
12. What is function overloading? Give an example

Ds Module

1. Basics

1. Explain linear and non-linear data structures.

2. Linked lists

1. Explain the types of linked lists WAP to reverse an SLL.
2. WAP to insert a node at the end of SLL.
3. WAP to remove duplicates in SLL.
4. WAP to delete the nth last element in an SLL.
5. What is the difference between algorithms and data structures?
6. Write code to find out whether the linked list is circular or linear.

3. Stack

1. What is a Stack? Explain with an example
2. Explain the complete code on the stack implementation and applications.
3. WAP to demonstrate push and pop operations in the stack.

4. Searching and Sorting Techniques

1. WAP to sort the array by using the bubble sort method
2. WAP to sort the given array by using merge sort
3. Explain quick sort algorithms.

4. Explain searching algorithms.

5. Queue

1. What is a Queue? Explain with an Example.
2. WAP to Implement a Queue using an array What is a circular queue?
3. WAP to implement a Circular queue.

6. Hashing

1. What is a hash table, and what is its use? Give an example of a hash table briefly about with an example. What is the time complexity of the hash program?

7. Trees

1. Explain the binary tree in detail.
2. Explain the binary search tree in detail.
3. WAP to print the pre-order output of a given BST
4. WAP to find the height of a tree.
5. WAP to print in-order output of a given BST.

LI Module

1. Basics

1. Explain the Linux commands like ls, mv, ifconfig, rm, and sed.
2. Difference between touch and vi command.
3. Explain the grep command in detail.
4. What is a file descriptor?
5. How to open the manual page of system calls?

2. System call

1. What is a system call? Explain with an example
2. Which is having the advantage, open or fopen?
3. Is an open system call OS independent?
4. Explain the difference between library calls and system calls.

3. Networking

1. Explain the difference between TCP and UDP.

2. What are the TCP and UDP protocols?

4. Process

1. What is the use of the fork system call?
2. Explain the use of the exec system call.
3. Explain how to create the process.
4. Explain the zombie and orphan process.

5. IPC

1. Explain the IPC mechanism in detail.
2. Explain the SHM in detail.
3. What is a pipe?
4. IPC mechanism brief Explanation (PIPE, FIFO, SHM).
5. What is the use of pipe(IPC), and what are the disadvantages?
6. Difference between pipe and FIFO.
7. Where to use pipe and FIFO?

6. Signal

1. What is the difference between signal and sigaction?
2. How many signals are we having?
3. Explain how to handle the SIGKILL signal.
4. Explain when the SIGCHLD signal will occur.
5. Print your name when the user presses Ctrl + C on the keyboard.

7. Socket

1. Explain socket programming.
2. In socket programming, have you used any system calls? List out if you used.

8. Threads

1. Explain the use of a thread.
2. What library will you use for multithread?
3. What is the difference between a mutex and a semaphore?
4. Explain the contact switching.
5. What is a semaphore?
6. Race condition? Critical section Implementation with mutex method and Semaphore (Using real-time example)

7. When we are using the mutex method and when we are using the semaphore. (with real-time example)
8. Instead of a mutex, if you use a semaphore? Instead of a semaphore, if you use a mutex, what happens? Explain in detail.
9. What is the difference between a binary mutex and a mutex with an example?
10. What is the difference between thread and pthread?
11. Explain about process and thread. Can we create threads or processes in Kernel Space?
12. What is the difference between a mutex and a semaphore?
13. What is the difference between a process and a thread?

9. Process and Memory Management

1. Explain the Process and memory management

General Questions

1. What do you know about ESP32?
2. Any work experience with STM32 microcontroller?