Questions.

**1. Sequa Technologies (Trivandrum)**

1. Introduce

2. Projects

3. What is meant by TDM?

4. Format of a Eth packet

5. Which microcontroller and its frequency

6. Mac address, and its significance in Packet forwarding.

When packets are routed, IP addresses are used to determine the next hop and the physical address is used to physically identify the interface serving as the next hop. Only the former (determining next-hop) is usually called routing. MAC addresses are discovered through ARP (Address Resolution Protocol) in IPv4 & ND6 (Neighbor Discovery) in IPv6.

The destination IP address in the IP header is the final destination. In the process of routing (at each hop), you get the next hop's IP address to (eventually) reach the final destination from the routing table (this could be a default gateway's IP address). To send the packet to the next hop, you need its MAC address. While hopping through intermediate links, the IP address in the IP header don't change - only the MAC addresses change.

7. Scheduling Algorithms, Critical Section, Preemption, Reentrancy.

* Scheduling algorithms :- mbed-os uses priority bases round robin scheduling
* Code inside a Critical section cannot be interrupted by any process. They will wait until the critical section is exit
* Preemption is pausing a task to serve higher priority task
* A function is said to be reentrant if it can be used by different threads at the same time.
* <https://en.wikipedia.org/wiki/Reentrancy_(computing)>
* <https://en.wikipedia.org/wiki/Thread_safety>

8. How to configure the priority of an interrupt.

9. How can u know the time taken for execution of a task.

Record times taken before and after executing the entire while loop in a task

10. I2C (full)

11. Is 7 a higher or lower priority intpt than 0.

In mbed-os it is lower priority

12. Static variables, scope.

13. How to define inside a structure.

14. #pragma and structure padding

15. How a stack overflow happen, explain with an example

16. Where are the function arguments stored.

**2. Quest Global**

1. Print fibbanocci.

2. Static and Dynamic Allocation difference

3. Print reverse of of number(most optimized)

4. How do you achieve task synchronization.

5. Difference between mutex and semaphore.

6. How many address supported by I2c.

7. difference between I2c and SPI.

8. How do you attain synchronization between interrupt and task.(Wait and Signal Mechanism)

9. Priority Inversion and Inheritance

10.Booting sequence of mc

11. Program to calculate angle between seconds and hour hand of a clock

12.Static variables and functions.

13. Predict the addressing of union declared in a structure.

14.IPC Mechanism in Linuxes.

15.Linked list and tree.

**3. Eximius Technologies**

1. Introduce

2. Projects

3. What is modular function

4. Difference between I2c and SPI

5. Which bit do you configure for read in SPI

6. tasks – scheduling.

7. What is Round Robin.

8. What is time slicing.

9. Semaphore and mutex

10. Static Keyword

11.Can Static be accessed externally in other files(told: Need to take global static variable to another global variable and then take extern, but he said no, take extern. Verified by programming taking extern of static variable is NOT Possible)

12. Difference between preprocessor and inline functiomn.

13. What is a volatile variable and where it is used(real life example)

14.DO you know device drivers, BSP

15.Constant Volatile

16.Dangling , Void Pointers

**4.HCL Technologies**

1.Introduce,Projects

2. Static function and scope.

3. Can two functions can static variables with same name(Yes)

4. Constant ,Volatile,Why they are used.

5. I/Pt latency.

6. RTOS ,scheduling.

7. Cooperative scheduling(Task Suspend)

8. Can macro be used instead of a constant.

9. GPIO pins alternate function.

10. UART

**5. Robert Bosch**

1.Memory Layout of a C and usage of different sections

2. Storage classes  
3.Volatile keyword.

4. Can const Volatile be declared.

5.Draw architecture of using microcontroller.

6. I2c explain.

7. How to configure OS\_Tick time in Free RTOS.

8. Set the ith bit of a byte.

9. Booting sequence .

10. Asked about A series of Arm.(not worked)

11. Write a program for dynamically storing a number of sting with varying string length. and we need to address or retrieve each stings independently.(double pointers)

12. Architecture of project Panasonic.

**Happiest Minds**

1. Introduce
2. Projects
3. What is tdm, which mc.
4. Deep and shallow copy of pointers.
5. I2C , SPI in detail(how many devices supported, which is better for bulk data communication)
6. If you 4 ss how many can u select without any external hw.
7. Cross/naïve compling.
8. DLL/ Static Linking
9. Static variables, Functions
10. Memory Layout of C
11. Where is ISR located,NVIC, Specific to microcontroller
12. Optimization Techniq
13. Stages of compilation
14. In which stage is the difference between cross and naïve compiler comes
15. Contents of .s file

**Wipro**

1. **Static, extern as globally.**
2. **Methodology used in UART communication.**
3. **Make a function to find the size of the input.**
4. **Pointers to the function.**
5. **Memory layout.**
6. **Inline functions.**
7. **Macro functions.**
8. **What is compiler optimization.**
9. **Volatile example program.**

**Quest**

1. **Interrupt latency**
2. **UART baud rate and bit rate.**
3. **Steps to read from a slave register I2C.**
4. **Use cases of const volatile in practical situation.**
5. **What will the linker be doing?**
6. **If there is only one main function, explain the steps of linker.**
7. **What is ISR.**
8. **How to pass a string to ISR.**
9. **How to return string from ISR.**
10. **Difference between normal function and ISR.**
11. **ISR steps.**
12. **When ISR execute where the values stored.**
13. **int i=0;**

**while(i>10) {**

**i—;**

**}**

**This program is written to make some delay. Calculations are correct, No error for program.**

**But delay time is different than expected. Why ?**

1. **Memory addressing of static variable if there using two static variable with same name.**
2. **Write a driver program for UART.**
3. **Static fuction.**