**Coding Questions**

1. Given a linked list of integers list, create a function that reverse it. Example 1: Input: list = 4 -> 8 -> 1 -> 6 -> 2 -> 5 -> null Output: 5->2→6→1→8→4→ null Embedded softwares: Embedded Software Ownership C/C++: Memory Management

2. #include int main() { int x = 10; int\* ptr1 = &x; int\*\* ptr2 = &ptr1; \*ptr1 = 20; \*ptr2 = nullptr; std::cout << x << std::endl; std::cout << \*ptr1 << std::endl; return 0; } Embedded softwares: Embedded Software Ownership C/C++: Memory Management Communication & Articulation: Communication & Articulation

**Theory/Scenario bases Questions**

1. #include int main() { int arr[] = {10, 20, 30, 40, 50}; int\* ptr1 = arr; int\* ptr2 = arr + 4; std::cout << (ptr2 - ptr1) << std::endl; // 1st part std::cout << \*(ptr1 + 3) << std::endl; // 2nd part std::cout << \*(ptr2--) << std::endl; // 3rd part std::cout << \*(++ptr1) << std::endl; // 4th part return 0; } Embedded softwares: Embedded Software Ownership C/C++: Memory Management Communication & Articulation: Communication & Articulation

2. You are given an IP address of 192.168.1.0 with a subnet mask of 255.255.255.192. How many subnets can be created from this configuration? Linux System: Networking Communication & Articulation: Communication & Articulation.

4. Have you used Git Framework for making branches and pushing the code CI/CD: Efficient Deployment Strategies.

5. Which Unit Test you used to test the code before pushing it to development? C/C++: Testing Frameworks.

6. #include void freeMemory(int\* ptr1, int\* ptr2) { delete ptr1; delete ptr2; } int main() { int\* p1 = new int; \*p1 = 100; int\* p2 = p1; freeMemory(p1, p2); std::cout << "Value of p1: " << \*p1 << std::endl; return 0; } C/C++: Memory Management

7. Given an array of integers arr and an integer k, create a boolean function that checks if there exist two elements in arr such that we get k when we add them together. Example 1: Input: arr = [4, 5, 1, -3, 6], k = 11 Output: true Explanation: 5 + 6 is equal to 11 Example 2: Input: arr = [4, 5, 1, -3, 6], k = -2 Output: true Explanation: 1 + (-3) is equal to -2 Example 3: Input: arr = [4, 5, 1, -3, 6], k = 8 Output: false Explanation: there is no pair that sums up to 8 Embedded softwares: Embedded Software Ownership

**Questions:**

1. You need to create a CI/CD pipeline that archives build artifacts for future use while ensuring version control and integrity. Implement this in YAML for a tool like GitHub Actions or Azure DevOps. CI/CD: Efficient Deployment Strategies

2. How would you set up alerts and notifications for critical errors in a CI/CD pipeline? CI/CD: Efficient Deployment Strategies.

3. Write a program that integrates with an API that enforces rate limiting. Implement a mechanism to respect the limits and retry requests after the cooldown period. C/C++: Third Party Libraries and API Integration.

4. Implement a C++ function to perform a POST request to an API endpoint with a JSON payload. Include bearer token authentication in the header. C/C++: Third Party Libraries and API Integration.

5. Write a program that simulates allocating and deallocating memory blocks of varying sizes. Analyze and optimize to reduce fragmentation, possibly using memory compaction techniques. C/C++: Memory Management.

6. A C++ application exhibits increasing memory usage over time. How would you investigate and debug a suspected memory leak caused by dynamic memory allocation? C/C++: Memory Management.

7. During a code review, you notice manual memory deallocation is inconsistently handled. What best practices would you recommend to ensure consistent and safe memory deallocation? C/C++: Memory Management

8. You are tasked with creating a CI/CD pipeline for a microservices application with 10+ services. How would you structure the pipeline to ensure faster deployments while keeping the process reliable? CI/CD: Efficient Deployment Strategies A C++ application exhibits increasing memory usage over time. How would you investigate and debug a suspected memory leak caused by dynamic memory allocation? C/C++: Memory Management You have a set of independent tasks that can be executed concurrently. Describe how you would implement a system to asynchronously execute these tasks using std::async and manage their results using std::future C/C++: Asynchronous Programming.

9. A C++ application exhibits increasing memory usage over time. How would you investigate and debug a suspected memory leak caused by dynamic memory allocation? C/C++: Memory Management.

10.You have a set of independent tasks that can be executed concurrently. Describe how you would implement a system to asynchronously execute these tasks using std::async and manage their results using std::future C/C++: Asynchronous Programming

11. If you have a log file that grows indefinitely, how would you write a shell script to rotate the log and keep only the last 5 logs? Linux System: Shell Scripting.

12. What is the difference between static routing and dynamic routing? When would you use each? Linux System: Networking.

13. If you are unable to access a server via SSH, but pinging the server works fine, how would you go about troubleshooting the issue on a Linux system? Linux System: Networking.

14. After deploying a new feature or system, how do you communicate the results and any lessons learned to the team? Communication & Articulation: Communication & Articulation.

15. When scaling a team for a high-performance project, what factors do you consider? How do you ensure that new team members integrate well and maintain team dynamics? Leadership and Engagement: High-Performance Team Building.

16. You notice that deployments in staging are failing due to an environment-specific configuration issue that wasn't present in the development pipeline. How would you update your CI/CD pipeline to handle this more effectively in the future? CI/CD: Efficient Deployment Strategies.

17. Write a program that simulates allocating and deallocating memory blocks of varying sizes. Analyze and optimize to reduce fragmentation, possibly using memory compaction techniques. C/C++: Memory Management.

18. Given a class hierarchy with virtual functions, demonstrate how polymorphism can impact memory management, especially in terms of object slicing and memory overhead from the vtable. How would you ensure proper memory management with polymorphic objects? C/C++: Memory Management.

19. Implement a program that computes the sum of elements in a large array using multiple threads. Divide the array into smaller chunks and process them concurrently, then combine the results at the end. C/C++: Asynchronous Programming.

20. Implement a C++ function to perform a POST request to an API endpoint with a JSON payload. Include bearer token authentication in the header. C/C++: Third Party Libraries and API Integration

21. Implement a retry mechanism for API calls that fail due to transient errors (e.g., HTTP 502). Limit retries to 3 attempts with exponential backoff. C/C++: Third Party Libraries and API Integration.

22. Write a program that dynamically allocates an array of integers, populates it with values, and then safely deallocates the memory using new and delete C/C++: Memory Management

23. Refactor the following code to use smart pointers (std::unique\_ptr, std::shared\_ptr, or std::weak\_ptr) for better memory management: class MyClass { private: int\* ptr; public: MyClass(int val) { ptr = new int(val); } ~MyClass() { delete ptr; } }; C/C++: Memory Management