Complete Roadmap for C Programming with Project.

-SolitaryHorkos

C Programming Roadmap – Advanced Level

🎯 Goals:

✅ Master advanced pointer concepts.
✅ Implement complex data structures (linked lists, trees, graphs).
✅ Optimize algorithms for efficiency.
✅ Work with advanced file handling and memory management.
✅ Learn multithreading and system-level programming.
✅ Gain experience with networking and socket programming.
✅ Develop robust debugging and error-handling techniques.

🔰 Phase 1: Advanced Pointers & Memory Management

📌 Advanced Pointer Techniques
🔹 Pointers to pointers (**ptr).
🔹 Function pointers and callbacks.
🔹 Void pointers (void *).

📌 Memory Management
🔹 Memory leaks and debugging tools (Valgrind).
🔹 Efficient memory allocation techniques.

📌 Bit Manipulation
🔹 Bitwise operations (&, |, ^, ~, <<, >>).
🔹 Bit fields in structures.

🔰 Phase 2: Complex Data Structures

📌 Linked Lists

🔹 Singly linked lists.

🔹 Doubly linked lists.

🔹 Circular linked lists.

📌 Stacks and Queues

🔹 Stack implementation using arrays and linked lists.

🔹 Queue and priority queue implementation.

📌 Trees & Binary Search Trees (BSTs)

🔹 Binary trees and BST implementation.

🔹 Tree traversal algorithms (Inorder, Preorder, Postorder).

📌 Graphs & Graph Algorithms

🔹 Graph representation (Adjacency list, adjacency matrix).

🔹 Depth-First Search (DFS) and Breadth-First Search (BFS).

🔹 Dijkstra’s Algorithm for shortest path finding.

📌 Implementing Custom Data Structures

🔹 Hash tables and their collision handling techniques.

🔹 Heaps (Min Heap, Max Heap) and their applications.

🔰 Phase 3: Algorithm Optimization & Advanced Sorting

📌 Sorting Algorithms

🔹 Merge Sort and Quick Sort.

🔹 Heap Sort and Counting Sort.

📌 Searching Algorithms

🔹 Linear search and Binary search.

🔹 Interpolation search.

📌 Algorithm Optimization

🔹 Complexity analysis (Big O notation).

🔹 Space-time trade-offs and optimizations.

🔰 Phase 4: Advanced File Handling

📌 Binary File Handling

🔹 Working with binary files (fwrite(), fread()).

🔹 File seeking (fseek(), ftell()).

📌 Memory-Mapped File Handling

🔹 Using mmap() for performance optimization.

🔰 Phase 5: Multithreading & System-Level Programming

📌 Multi-threading (POSIX Threads – Pthreads)

🔹 Introduction to the POSIX pthread library.

🔹 Thread creation and management.

🔹 Synchronization mechanisms (mutexes, semaphores).

🔹 Deadlock prevention and detection.

📌 Interprocess Communication (IPC)

🔹 Pipes and shared memory.

🔹 Message queues and semaphores.

📌 Bitwise Operations for Optimization

🔹 Efficient data representation and bit masking.

🔹 Using bitwise operations for performance gains.

Complete Roadmap for C Programming with Project.

-SolitaryHorkos

🔰 Phase 6: Networking & Socket Programming

📌 Socket Programming in C

🔹 Understanding TCP/IP networking.

🔹 Implementing client-server models using sockets.

🔹 Creating simple network applications.

🔰 Phase 7: Preprocessor Directives & Advanced Debugging

📌 Preprocessor Directives

🔹 Writing macros with arguments.

🔹 Conditional compilation using #ifdef, #ifndef.

🔹 Smart macros and compile-time optimizations.

📌 Command-Line Arguments

🔹 Using argc and argv for command-line input handling.

📌 Advanced Debugging Techniques

🔹 Debugging with gdb.

🔹 Memory profiling with Valgrind.

🔹 Handling errno and perror for error diagnostics.

📌 Error Handling

🔹 Implementing custom error handling techniques.

🔹 Defensive programming practices.

🏆 Skills to Master by the End of This Phase:

✅ Implementing complex data structures and algorithms.

✅ Writing highly optimized and efficient code.

Complete Roadmap for C Programming with Project.

-SolitaryHorkos

✅ Handling advanced file operations and memory management.
✅ Working with multithreading and synchronization.
✅ Building network-based applications using socket programming.
✅ Debugging and profiling applications effectively.

🎯 PROJECTS:

- **Chess Game** – A 2-player chess game using data structures.
- **Tetris Game** – Implement the classic Tetris game in C.
- **Minesweeper (CLI)** – A command-line version of Minesweeper.
- **Snake Game** – Create a simple 2D Snake game using C.
- **Sudoku Solver** – Solve Sudoku puzzles using backtracking.
- **Maze Solver** – Implement pathfinding algorithms like BFS/DFS.
- **Game using SDL** – Develop a simple graphical game using the SDL library.
- **Chat Application** – Real-time chat using sockets (TCP/UDP).
- **HTTP Server in C** – Develop a basic web server with socket programming.
- **Multi-threaded Web Server** – Handle concurrent requests using threads.
- **Network Packet Sniffer** – Capture and analyze network packets.
- **Process Scheduler Simulator** – Simulate CPU scheduling algorithms (e.g., Round-Robin, FCFS).
- **Memory Allocator** – Implement a custom memory management system like malloc/free.
- **Simple Process Manager** – List and terminate system processes.
- **File Compression Utility** – Implement Huffman coding for compression.

Complete Roadmap for C Programming with Project.

-SolitaryHorkos

- **Basic File Compression/Decompression** – Apply simple compression algorithms.
- **Simple Database using Files** – Store and retrieve structured data using file handling.
- **Simple Compiler or Interpreter** – Implement a basic compiler for a small language.
- **Enhanced Unix Shell** – Support piping, redirection, and job control.
- **Simple Shell** – A basic command-line shell with command execution.
- **Binary Search Tree (BST)** – Implement insertion, deletion, and traversal.
- **Linked List Implementation** – Perform various linked list operations.
- **Password Manager** – Securely store and retrieve passwords.
- **Encryption/Decryption Tool** – Implement simple encryption techniques (Caesar cipher, XOR, etc.).
- **Graphical Calculator** – A GUI-based calculator using GTK or similar.
- **Command-Line To-Do List** – Manage tasks via CLI.
- **Hospital Management System** – Manage patient records efficiently.
- **Online Quiz System** – A multiple-choice quiz with scoring.
- **Music Player** – Play audio files using C libraries.