C Programming: A Roadmap

1. Introduction

* History of C
* Importance of C
* C Programming Language Standard
* Basic Skeleton of a C Program

2. Fundamentals

* The basic structure of C programming
* How C programming works
* How a C compiler works
* What is a linker and how it works
* What happens in computer memory
* What is an executable file
* Using terminal to run your very first program

3. Variables and Data Types

* What is a variable
* Declaring variables and naming conventions
* Keywords and identifiers
* Static variable vs global variable
* Basic number systems
* ASCII Table
* Basic input-output functions (printf, scanf)
* Escape sequences
* Placeholders / conversion specifiers
* Formatted input and output
* Fundamental data types and valid ranges
* size_t data type in C
* How to typecast in C

4. Operators

* Arithmetic Operators

* Relational Operators
* Logical Operators
* Assignment Operators
* Increment and Decrement operators
* Conditional operators
* Bitwise Operators
* Special operators
* Precedence and Associativity of operators

5. Control Flow

* Software development method (SDM)
* Specification of needs
* Problem Analysis
* Design and Algorithms representation
* Implementation
* Testing and Verification
* Documentation
* Selection Structures (if, if-else, else if, nested if)
* Programming style/ Indentation style
* Short Circuit Evaluation
* Multiple Selection style (switch-case)
* Data Representation using the signed magnitude
* Repetition Structures (while, do-while, for, infinite loop)
* Loop Control statements (break, continue)
* Nested loops
* Exercise: Solve at least 20 pyramids of stars program using loops

6. Arrays

* Introduction to Arrays
* Definition, Declaration, Initialization, and Accessing Array elements
* Designated initialization of arrays
* Multi-dimensional arrays

* Exercise: Solve at least 30 problems using 1D arrays and at least matrix adding and multiplying problems using 2D arrays

7. Searching and Sorting

* Searching and Sorting in Arrays (Bubble sort, Selection sort, Insertion sort, Linear search, Binary search)

8. Strings

* Introduction to Character array and strings

* Declare and initialize string variables

* String handling libraries (ctype.h, string.h, stdlib.h)

9. Functions

* Function Declaration and Definition

* Difference between Parameters and Arguments

* Call by value and call by reference

* Static and Dynamic scoping

* Heap and Stack concepts

* Introduction to Recursion

* Types of Recursion (direct, indirect, tail, and non-tail)

* Advantages and Disadvantages of recursion

* Exercise: Solve minimum 10 problems using recursion

* Passing Arrays to functions

* Passing Strings to functions

* Scope, visibility, and lifetime of variables

* Automatic variables, external variables, static variables, register variables

10. Pointers

* Introduction to Pointers

* Definition, Declaration, Initialization, and Accessing pointers

* Value of operator in pointers

* Arrays and Pointers

* Pointers and character strings

* Arrays of pointers

* Pointer as function arguments

* Pointer to functions

* Benefits of pointers
* Pointer and functions
* Using pointers to 2D Arrays
* String literals
* Dynamic Memory Allocation (malloc, calloc, realloc, free)

11. Data Structures

* Concept of linked lists
* Application of linked lists
* Linked list vs array
* Structures and Unions in C
* Declaring and Accessing structure members
* Array of structures
* Structures and functions

12. Preprocessor

* Introduction to the Preprocessor
* Macro substitution, file inclusion, compiler control directives

13. File Handling

* File management in C
* File and Stream
* Define, open, and closing a file
* Input-output operations on files
* Error handling during I/O operations

14. Input/Output

* Command line arguments

15. Debugging and Testing

* Common programming errors
* Program testing and debugging

16. Projects

* Library management project
* Calendar application
* Student information project

* Office management system

This outline provides a structured roadmap for your C programming journey. Remember that consistent practice and problem-solving are crucial for effective learning.

I hope this helps! Let me know if you have any other questions.