Here are MCQs, Short and Long Q&A questions based on the topics in the image you sent:

Computer Science XII - C Language Exam Questions

Note: These questions are designed based on the syllabus provided in the image.

I. Multiple Choice Questions (MCQs)

Topic 1: Introduction and history of C language

1. **Which decade was C language developed in?**
   a) 1960s
   b) 1970s
   c) 1980s
   d) 1990s
   Answer: b) 1970s
2. **C language was developed at:**
   a) Microsoft Corporation
   b) Google
   c) Bell Laboratories
   d) IBM
   Answer: c) Bell Laboratories
3. **C is considered a ______ level language.**
   a) High
   b) Low
   c) Middle
   d) Machine
   Answer: c) Middle

Topic 2: Basic Structure/Block structure of C program

1. **Which stage is the preprocessor directive processed in?**
   a) Compilation
   b) Linking
   c) Execution
   d) Preprocessing
   Answer: d) Preprocessing
2. **What is the entry point of a C program?**
   a) start()
   b) main()
   c) program()
   d) init()
   Answer: b) main()
3. **Which symbols are used to enclose the body of the main() function?**
   a) ()
   b) {}
   c) []
   d) <>
   Answer: b) {}

Topic 3: Explain (Variables, Identifiers, Token, Constant)

1. **Which of the following is NOT a valid identifier in C?**
   a) _variable
   b) variable_1
   c) 1variable
   d) variable
   Answer: c) 1variable
2. **A keyword in C is a:**
   a) User-defined word

b) Predefined word with special meaning
c) Variable name
d) Constant
Answer: b) Predefined word with special meaning

3. **Which of the following is a constant in C?**
   a) int age;
   b) const float PI = 3.14;
   c) char name[20];
   d) void function();
   Answer: b) const float PI = 3.14;

Topic 4: Data types with sizes

1. **What is the size of an int data type in a typical 32-bit C compiler?**
   a) 1 byte
   b) 2 bytes
   c) 4 bytes
   d) 8 bytes
   Answer: c) 4 bytes (Note: Size can vary depending on the compiler and architecture)
2. **Which data type is used to store character values?**
   a) int
   b) float
   c) char
   d) string
   Answer: c) char
3. **Which data type is used to store floating-point numbers?**
   a) int
   b) char
   c) double
   d) string
   Answer: c) double (or float)

Topic 5: Operators and its usages

1. **Which operator is used for assignment in C?**
   a) ==
   b) =
   c) +
   d) *
   Answer: b) =
2. **Which operator is used for logical AND?**
   a) &
   b) |
   c) &&
   d) ||
   Answer: c) &&
3. **What is the operator % used for in C?**
   a) Division
   b) Modulus (remainder)
   c) Percentage
   d) Multiplication
   Answer: b) Modulus (remainder)

Topic 6: Control structures - Selection/Decision making Structure

1. **Which control structure is used for decision making in C?**
   a) for loop
   b) while loop
   c) if-else statement

d) do-while loop
Answer: c) if-else statement

2. **Which statement is used to execute one block of code from multiple blocks based on a condition?**
   a) for
   b) while
   c) if-else if-else
   d) do-while
   Answer: c) if-else if-else
3. **Which statement allows you to choose one option from several choices?**
   a) if
   b) switch
   c) while
   d) for
   Answer: b) switch

Topic 6: Control structures - Iterative Structure

1. **Which loop is an entry-controlled loop?**
   a) do-while
   b) for
   c) while
   d) Both for and while
   Answer: d) Both for and while
2. **Which loop is an exit-controlled loop?**
   a) for
   b) while
   c) do-while
   d) if
   Answer: c) do-while

Topic 6: Control structures - Unconditional

1. **Which keyword is used to terminate the loop or switch statement immediately?**
   a) continue
   b) return
   c) break
   d) goto
   Answer: c) break
2. **Which keyword is used to skip the current iteration and proceed to the next iteration of a loop?**
   a) break
   b) return
   c) continue
   d) goto
   Answer: c) continue
3. **Which keyword is used to transfer control to a labeled statement?**
   a) break
   b) return
   c) continue
   d) goto
   Answer: d) goto

Topic 7: Functions - System define/Predefine function

1. **printf() is an example of a:**
   a) User-defined function
   b) System-defined function
   c) Main function

d) Recursive function
Answer: b) System-defined function

2. **Which header file is required to use the printf() function?**
   a) stdio.h
   b) conio.h
   c) math.h
   d) string.h
   Answer: a) stdio.h

Topic 7: Functions - User define functions

1. **What is the process of creating a function called?**
   a) Function calling
   b) Function declaration
   c) Function definition
   d) Function prototype
   Answer: c) Function definition
2. **What is the process of invoking a function called?**
   a) Function definition
   b) Function declaration
   c) Function calling
   d) Function prototype
   Answer: c) Function calling

Topic 8: Array and Strings - Array declaration, Initialization, 2D Array

1. **How do you declare an integer array of size 5 named numbers?**
   a) int numbers;
   b) int numbers[5];
   c) array numbers[5];
   d) int array numbers[5];
   Answer: b) int numbers[5];
2. **How do you initialize an integer array arr of size 3 with values 1, 2, and 3?**
   a) int arr = {1, 2, 3};
   b) int arr[] = {1, 2, 3};
   c) int arr[3] = {1, 2, 3};
   d) All of the above
   Answer: d) All of the above
3. **How do you declare a 2D array of integers with 3 rows and 4 columns?**
   a) int matrix[3][4];
   b) int matrix[3, 4];
   c) int matrix[][];
   d) int matrix[4][3];
   Answer: a) int matrix[3][4];

Topic 8: Array and Strings - String and few function of string

1. **In C, a string is terminated by a:**
   a) Space character
   b) Null character \0
   c) Newline character \n
   d) End of file
   Answer: b) Null character \0
2. **Which function is used to read a string with spaces from the user?**
   a) scanf()
   b) printf()
   c) gets()
   d) puts()
   Answer: c) gets()

3. **Which function is used to copy one string to another?**
   a) strcat()
   b) strcmp()
   c) strcpy()
   d) strlen()
   Answer: c) strcpy()
4. **Which function is used to compare two strings?**
   a) strcat()
   b) strcmp()
   c) strcpy()
   d) strlen()
   Answer: b) strcmp()
5. **Which function is used to concatenate (join) two strings?**
   a) strcat()
   b) strcmp()
   c) strcpy()
   d) strlen()
   Answer: a) strcat()

Topic 9: Pointers

1. **What does a pointer variable store?**
   a) Value of a variable
   b) Data type of a variable
   c) Address of a variable
   d) Name of a variable
   Answer: c) Address of a variable
2. **Which operator is used to get the address of a variable?**
   a) *
   b) &
   c) ->
   d) .
   Answer: b) &
3. **Which operator is used to access the value at the address pointed to by a pointer?**
   a) &
   b) *
   c) ->
   d) .
   Answer: b) *

II. Short Question & Answers

Topic 1: Introduction and history of C language

1. **What are the key features of the C language?**
   - *Answer:* Portability, efficiency, mid-level language, structured programming language, rich set of built-in functions, extensibility.
2. **Name at least two languages that were influenced by C.**
   - *Answer:* C++, Java, C#, Python, Perl, etc. (Any two will suffice)

Topic 2: Basic Structure/Block structure of C program

1. **Explain the three stages of compiling a C program.**
   - *Answer:* Preprocessing (handling directives), Compilation (source code to assembly code), Linking (combining object code and libraries to create executable).
2. **What is the role of header files in a C program? Give an example.**
   - *Answer:* Header files contain function declarations, macro definitions, and other pre-written code that can be included in your program. Example: stdio.h for standard input/output functions.

Topic 3: Explain (Variables, Identifiers, Token, Constant)

1. **Differentiate between variables and constants in C.**
    - *Answer:* Variables are named storage locations that can hold values that can be changed during program execution. Constants are fixed values that cannot be changed during program execution.
2. **What are tokens in C? List different types of tokens.**
    - *Answer:* Tokens are the smallest individual units in a C program. Types: Keywords, Identifiers, Constants, Strings, Special symbols, Operators.

Topic 4: Data types with sizes

1. **List the basic data types in C along with their typical sizes.**
    - *Answer:* int (typically 4 bytes), char (1 byte), float (4 bytes), double (8 bytes), void (0 bytes). (Sizes may vary based on the system).
2. **What is the difference between float and double data types?**
    - *Answer:* Both are for floating-point numbers. double provides higher precision and a larger range than float. double typically uses 8 bytes, while float uses 4 bytes.

Topic 5: Operators and its usages

1. **Explain the difference between logical AND (&&) and bitwise AND (&) operators.**
    - *Answer:* Logical AND (&&) operates on boolean values (true/false) and returns true only if both operands are true. Bitwise AND (&) operates on individual bits of integers.
2. **List and briefly explain any three types of operators in C.**
    - *Answer:*
        - **Arithmetic Operators:** (+, -, *, /, %) for mathematical operations.
        - **Relational Operators:** (==, !=, >, <, >=, <=) for comparisons.
        - **Logical Operators:** (&&, ||, !) for logical operations.
        - **Assignment Operators:** (=, +=, -=, *=, /=, %=) for assigning values.
        - **Increment/Decrement Operators:** (++ , --) to increase or decrease value by 1.

Topic 6: Control structures - Selection/Decision making Structure

1. **Explain the if-else if-else ladder with a simple example.**
    - *Answer:* Used for multi-way decision making. It checks conditions sequentially, and executes the block associated with the first true condition. The else block (optional) executes if none of the conditions are true. (Provide a simple code example demonstrating its use).
2. **When is the switch statement preferred over if-else if-else?**
    - *Answer:* When you need to choose one option from a set of discrete values for a single variable. switch is more efficient and readable for such cases.

Topic 6: Control structures - Iterative Structure

1. **Compare and contrast while and do-while loops.**
    - *Answer:* while loop is entry-controlled (condition checked before execution), do-while loop is exit-controlled (condition checked after execution, loop executes at least once).
2. **Explain the syntax of a for loop and when it is typically used.**
    - *Answer:* Syntax: for (initialization; condition; increment/decrement) { // loop body }. Used when you know the number of iterations in advance or need to iterate a specific number of times.

Topic 6: Control structures - Unconditional

1. **Explain the use of break and continue statements within loops.**

    - *Answer:* break terminates the loop execution immediately and transfers control to the statement after the loop. continue skips the current iteration of the loop and proceeds to the next iteration.

Topic 7: Functions - System define/Predefine function

1. **What are system-defined (predefined) functions? Give two examples.**
    - *Answer:* Functions that are built into the C library and are readily available for use. Examples: printf(), scanf(), sqrt(), pow(), strlen(), etc.

Topic 7: Functions - User define functions

1. **Explain the concept of function declaration, definition, and calling.**
    - *Answer:*
        - **Declaration:** Informs the compiler about the function's name, return type, and parameters (prototype).
        - **Definition:** Contains the actual code or body of the function.
        - **Calling:** Invoking the function from another part of the program to execute its code.
2. **What are the advantages of using user-defined functions?**
    - *Answer:* Code reusability, modularity, improved readability, easier debugging, abstraction, and breaking down complex problems into smaller, manageable parts.

Topic 8: Array and Strings - Array declaration, Initialization, 2D Array

1. **What is an array? How is it declared and initialized in C?**
    - *Answer:* An array is a collection of elements of the same data type stored in contiguous memory locations. Declaration: dataType arrayName[arraySize]; Initialization: dataType arrayName[arraySize] = {value1, value2, ...};
2. **Explain the concept of a 2D array with a simple example.**
    - *Answer:* A 2D array is an array of arrays, organized in rows and columns. It's used to represent tables or matrices. Example: int matrix[3][3]; (Explain how to access elements using row and column indices).

Topic 8: Array and Strings - String and few function of string

1. **What is a string in C? How is it represented?**
    - *Answer:* A string in C is an array of characters terminated by a null character (\0). Represented as char stringName[size];
2. **Explain the purpose and syntax of strcpy(), strcat(), and strlen() functions.**
    - *Answer:*
        - strcpy(destination, source);: Copies string source to destination.
        - strcat(destination, source);: Appends string source to the end of destination.
        - strlen(string);: Returns the length of the string (excluding the null character). (Explain syntax and usage with brief examples).

Topic 9: Pointers

1. **What is a pointer? How is it declared and initialized?**
    - *Answer:* A pointer is a variable that stores the memory address of another variable. Declaration: dataType *pointerName; Initialization: pointerName = &variableName; (Explain the * and & operators).
2. **Explain the benefits of using pointers in C.**
    - *Answer:* Dynamic memory allocation, efficient data manipulation (especially with arrays and strings), passing arguments by reference in functions, implementing data structures like linked lists and trees.

III. Long Question & Answers

Topic 2: Basic Structure/Block structure of C program

1. **Describe the basic structure of a C program with a detailed explanation of each section. Include preprocessor directives, the main() function, and the importance of delimiters/braces.**
    - *Answer:* (Provide a detailed explanation of each section of a C program structure. Include:
        - **Preprocessor Directives:** Explanation of #include, #define, etc., and their role in preprocessing.
        - **Global Declarations:** Variables declared outside main().
        - **main() function:** Entry point, return type, function body.
        - **Local Declarations:** Variables declared inside main().
        - **Executable statements:** Code within main().
        - **Delimiters/Braces {}:** Importance in defining code blocks and function bodies.
        - Illustrate with a simple example C program and label each section.)

Topic 6: Control structures

1. **Discuss different types of control structures in C, explaining their purpose, syntax, and providing suitable examples for each (Selection, Iteration, and Unconditional).**
    - *Answer:* (Explain each category of control structures with details and examples:
        - **Selection/Decision Making:** if, if-else, if-else if-else, switch-case. Explain syntax, flow, and provide code examples for each.
        - **Iteration/Looping:** for, while, do-while. Explain syntax, entry vs. exit control, when to use each, and provide code examples for each.
        - **Unconditional Control:** break, continue, goto, return. Explain their purpose and usage within control flow with examples.)

Topic 7: Functions

1. **Explain the concept of functions in C. Differentiate between system-defined and user-defined functions. Describe the steps involved in creating and using user-defined functions (declaration, definition, calling). Discuss the different methods of user-defined functions (No arguments no return, arguments no return, no arguments with return, arguments with return) with illustrative examples for each method.**
    - *Answer:* (Comprehensive explanation of functions:
        - **Introduction to Functions:** Definition, purpose, advantages.
        - **System-defined vs. User-defined:** Difference and examples of each.
        - **User-defined Function Steps:** Detailed explanation of declaration (prototype), definition (function body), and calling (function invocation).
        - **Methods of User-defined Functions:** Explain and provide code examples for each of the four methods:
            - No arguments and no return value (void return type, void parameters).
            - Arguments but no return value (void return type, parameters).
            - No arguments but with return value (return type, void parameters, return statement).
            - Arguments and with return value (return type, parameters, return statement).

Topic 8: Array and Strings

1. **Explain arrays and strings in C. Describe how to declare, initialize, and access array elements, including 2D arrays. Discuss common string manipulation functions like gets(), puts(), strlen(), strcpy(), strcat(), and strcmp() with syntax and examples of their usage.**
    - *Answer:* (Detailed explanation of arrays and strings:

        - **Arrays:** Definition, declaration, initialization (different ways), accessing elements using indices, concept of contiguous memory, advantages of arrays.
        - **2D Arrays:** Declaration, initialization, accessing elements using row and column indices, representation as matrices, examples of use cases.
        - **Strings:** Definition, representation as char arrays, null termination, declaration, initialization.
        - **String Manipulation Functions:** Explain each function listed: gets(), puts(), strlen(), strcpy(), strcat(), strcmp(). For each function, provide:
            - Purpose and functionality.
            - Syntax.
            - Example code demonstrating its use.)

Topic 9: Pointers

1. **Explain pointers in C with examples. Discuss pointer declaration, initialization, and dereferencing. Explain how pointers are related to arrays and demonstrate pointer arithmetic. Discuss the advantages and potential pitfalls of using pointers.**
    - *Answer:* (In-depth explanation of pointers:
        - **Introduction to Pointers:** Definition, purpose (memory addresses), need for pointers.
        - **Pointer Declaration and Initialization:** Syntax, data type association, initialization with & operator, null pointers.
        - **Dereferencing Pointers:** Using the * operator to access the value pointed to by a pointer.
        - **Pointers and Arrays:** Relationship between array names and pointers (array name as a pointer to the first element), accessing array elements using pointer arithmetic.
        - **Pointer Arithmetic:** Explain addition, subtraction, increment, decrement of pointers and how it relates to data type size.
        - **Advantages of Pointers:** Dynamic memory allocation, efficiency, function arguments by reference, data structures.
        - **Pitfalls of Pointers:** Dangling pointers, memory leaks, segmentation faults, importance of careful memory management.)

These questions cover the syllabus provided and are designed to test understanding at different levels. Remember to study each topic thoroughly and practice coding to be well-prepared for your exam. Good luck!