**Module 10: Programming an SoC Using C Language**

1. Which of the following statements is correct?

1. The use of C language facilitates the handling of complex data structures and allows the programmer to have direct control over instruction sequence generation.
2. C language is portable, but it does not support access to core registers and stacks.
3. C language facilitates the handling of complex data structures but does not allow the programmer to have direct control over instruction sequence generation.
4. C language is portable and typically allows the programmer to have full control over instruction sequence generation.

2. Which of the following statements is correct?

1. The use of assembly language facilitates the handling of complex data structures and allows the programmer to have direct control over instruction sequence generation.
2. Assembly language is portable, but does not support access to core registers and stacks.
3. Assembly language facilitates the handling of complex data structures but it does not allow the programmer to have direct control over instruction sequence generation.
4. Although assembly language is typically less portable than C, it allows the programmer to have full control over instruction sequence generation.

3. Which of the following tasks is performed by the assembler?

1. Converting C code into a sequence of assembly language instructions
2. Translating assembly codes into machine codes
3. Linking the object files generated by the compiler with the object code files for any library functions used in the program to produce an executable file
4. All of the above.

4. Which of the following tasks is performed by the compiler?

1. Converting an assembly code into a sequence of C language instructions
2. Translating assembly codes into machine codes
3. Linking the object files generated by the compiler with the object code files for any library functions used in the program to produce an executable file
4. None of the above.

5. Which of the following files is likely to have been generated by the linking stage of a compilation process?

1. code.axf
2. code.o
3. code.hex
4. code.s

6. Which part of the program image in the memory normally contains the starting addresses of exceptions?

1. Start-up routine
2. Program code
3. Vector table
4. C library code

7. Which C function is likely to be included in the C library part of a program image to be executed on a Cortex-M0 processor?

1. Fixed point multiplication
2. Floating point multiplication
3. Fixed point division
4. Floating point division

8. Which information is likely to be included in the start-up routine part of a program image to be executed on a Cortex-M0 processor?

1. Initial values of local variables within functions and subroutines
2. Addresses of peripherals
3. Instructions generated from the application program
4. Values for global data variables

9. Program data are normally stored in volatile memory; which of the following is normally held in the heap section of the program data?

1. Global variables and static variables
2. Pieces of memory spaces dynamically reserved by function calls
3. Temporary data for local variables; parameters passing in function calls
4. Instructions generated from the application program

10. Which of the following registers may need to be stored in the stack when a C function is called from an assembly file? (There may be more than one correct answer.)

1. R0 and R1
2. R2 and R3
3. R12
4. R14

11. Which of the following objects should be defined using the volatile class qualifier in a C program?

1. Constant data objects stored in volatile memory
2. Constant data objects stored in non-volatile memory
3. A data object that can be modified by an interrupt service routine
4. None of the above.