QUESTION 1  
  
Meets Expectation  
  
Feedback: Great job on your explanation of the Hardware Abstraction Layer (HAL). You correctly identified what HAL is and its purpose in simplifying interaction with the hardware. You also provided a good example of how HAL can be used in a project, which shows your understanding of its practical application. For future work, consider expanding your explanation to include more details about why it is useful to have a HAL in an embedded project. For example, you could discuss how it can increase code portability or reduce development time.  
  
QUESTION 2  
  
Does not meet expectation.  
  
Your explanation of what the \_\_io\_putchar function does is correct, but you missed out on explaining why it is needed. Remember, this function is not just about sending characters to an output channel, but it's also about enabling printing to the UART or console. It's used by the printf function and is required because different projects may require different hardware for printing. Try to include these points in your explanation to fully answer the question.  
  
QUESTION 3  
  
Meets Expectation  
  
Feedback: You've done a good job explaining why the debugger was necessary for this lab. You correctly identified that the debugger allows us to examine the values stored in the registers and create breakpoints in the code. This is a crucial aspect of systems programming on the Cortex M4 microcontroller. Keep up the good work! For future reports, consider expanding on why being able to create breakpoints and examine registers at specific points in the code is beneficial.  
  
QUESTION 4  
  
Does not meet expectation.  
  
Feedback: You've done a good job explaining how the new value for PSP is computed by offsetting the value of MSP\_init\_val. You've correctly identified that the stack counts down and that we need to skip past the main stack to get to the first thread's stack. However, you could improve your answer by explaining why we need to know the initial location of MSP for this to work. Remember, understanding the initial state of the system is crucial for ensuring that our value for PSP will be a valid stack address.  
  
QUESTION 5  
  
Meets Expectation  
  
Feedback: Excellent work! You've correctly identified that the start of each stack is a constant number of bytes away from the start of the last stack. Your understanding of how to find the nth thread stack's starting address by subtracting n times the size of each stack from the MSP initial value is spot on. Keep up the good work! In future, you might want to consider discussing how to ensure that the new address remains a valid stack address.