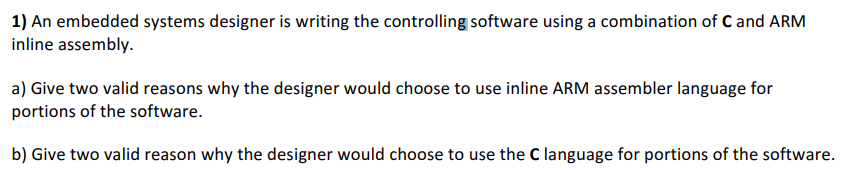
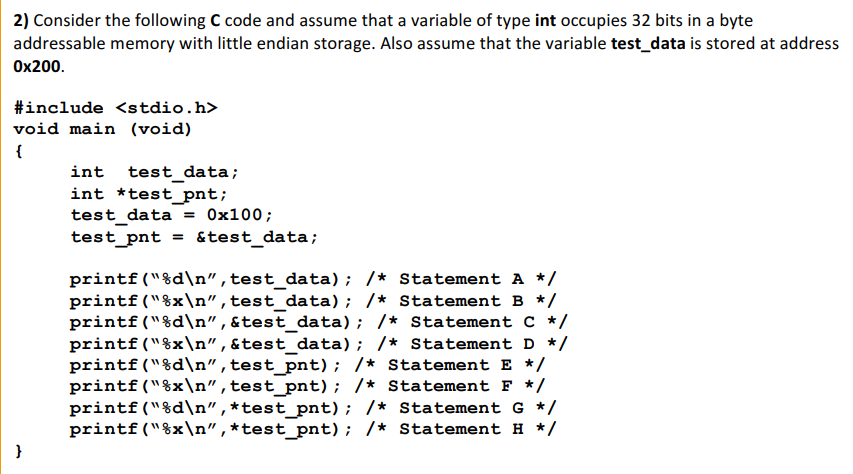
Microcontroller Architecture and interfacing homework 3



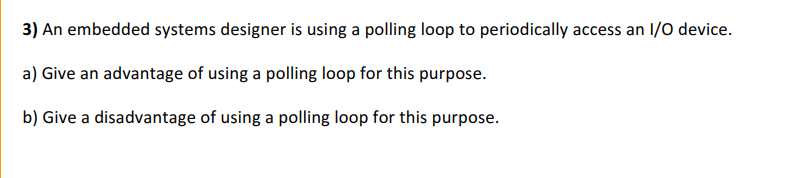
* to reduce power consumption
* To reduce memory usage/foot print



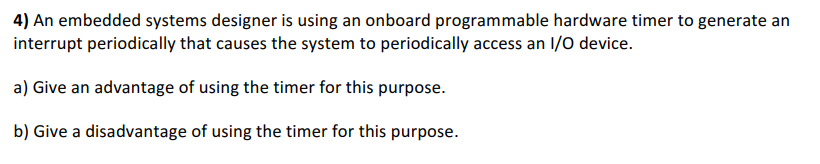
* To have a code that is easier to read and understand thereby easier to work on with a team
* To call on functions from libraries such as math.h to solve arithmetic operations(much harder/annoying to do on ARM)



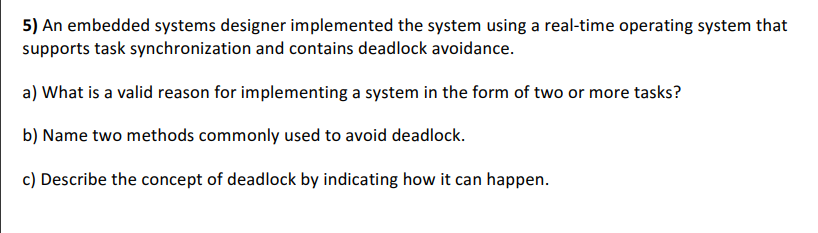
1. 256
2. 0x100
3. 512
4. 0x200
5. 512
6. 0x200
7. 256
8. 0x100



1. No interrupts and no timers are used therefore you don’t need to worry about priorities of each task
2. The processor is stuck doing this same task until it is finished so it can not be working on another task further down the pipeline



1. Lets the processor perform other things and multitask while checking for any change in the I/O ports
2. Requires initializing the timer and interrupts which need to be used at the right times and not excessively.



1. So that each function can be viewed as one task and let the operating system sort out a time schedule for which task to execute at what time
2. MUTEX and Semaphores
3. When there is too many programs running simultaneously and there is not enough resources/computing power to process each programs needs and there is not priority given to each task that needs to be done causing all the tasks to need something to complete themselves