Solutions 9

Question 1: (9 marks)

a) struct SomeStruct *foobar = (struct SomeStruct *)0x20000400;   // typecase to prevent warning (2)

b) 0x2000 0400 + 4 + (8*2) + 4 + 2 = 0x2000 041A and 0x2000 041B (2)

c) foobar->d[1] = 0; (1)

d) A #define allows one to give some code fragment (aka: macro) a name. Wherever that name is used the preprocessor simply replaces the name with the code fragment. In the ST header file #defines are used to map peripheral names to the base addresses of the peripherals typecast to structs. (2)

e) When the preprocessor encounters a #include directive, it reads the entire contents of the included file as if it had been typed out at the location of the #include (2)

Question 2: (14 marks)

a) The Virtual Memory Address is the location where the contents of the section will be accessed at runtime. The Load Memory Address is where the initial values of the section will be loaded into memory. (3)

b) The section containing statically allocated values has a LMA in flash and a VMA in RAM. Hence, the initialisation values are permanently placed into flash. On reset, in the startup code, the initialization values are copied from LMA (flash) to VMA (RAM). (2)

c) It's common for statically allocated variables to be initialised to 0. By pulling all variables initialised to 0 into a block (the .bss section), we need only store the start and end address of that block and can then easily set everything in that block to 0. This is more efficient than storing a whole block of 0s in flash. (2)

d) If static is placed in front of a function name or global variable, it makes the function or variable statically visible, or only visible within the file it's defined in. If static is placed in front of a local variable it makes that variable statically allocated thereby having a fixed space in memory. A local variable will always only be visible to the function it's defined in whether statically allocated or automatically allocated.
Static visibility should be implemented because having a name (whether it's a function name or variable name) visible only where it's needed prevents that name from interfering or accidentally being used elsewhere in the project. (5)

e) When information needs to be shared between your main thread and an ISR. Reason: an ISR is not called by the programmer, so it's not possible to pass in arguments. (2)