Solutions 8

Question1

a) a pointer to an 8-bit signed int or int8_t * {{1}}

b) Datatype: uint32_t * or pointer to 32-bits of unsigned data. {{2}}
Value: 0xAABBCCDD + (4 * 17) = 0xAABBCD21

c)
0x2000 0510 + (32 * 4) = 0x2000 0590. {{4}}

Therefore:
0x20000590: 0xAA
0x20000591: 0x00
0x20000592: 0x00
0x20000593: 0x00

Question 2 {{3}}

The address of foo is passed in as ‘a’, the address of bar is passed in as ‘b’
The data pointed to by a ‘a’ (foo) is set to the data pointed to by ‘a’ (foo) plus the data pointed to by ‘b’ (bar) = 0xAABB + 0xCCDD = 0x17798.
(1 mark)

However, foo is a 16-bit number so the value is truncated to 0x7798
(1 mark)

b++ modifies the *local* variable ‘b’, and hence has no effect on the data being passed in
(1 mark)

Bonus: {{2}}

When foo and bar are defined, foo is defined first on the stack (higher address) and bar is defined second (lower address).
By incrementing the variable holding the address of bar, the pointer ‘b’ is incremented to point to the higher address, namely ‘foo’.
Hence, both pointers would point to foo, so the computation would be
foo = 0xAABB + 0xAABB = 0x15576
Truncated: 0x5576

To get the marks, they must point out the order which the variables are defined on the stack and how foo has a higher memory address.