Solutions 8

Question 1: (10 marks)

a) 4 bytes. Reason: the value of a pointer is the memory address of what it is pointing to. Memory addresses are 32 bits wide. 2 marks

b) Yes. There is no universal fixed size for an address. A memory space can be implemented with whatever address space the designer wishes. Popular address spaces have included 16 bit, 32 bit and in the last few years 64 bit. 2 marks

c) Reference: gets the effective address of a variable. Ie: where that variable has been placed in memory. Defined on all variables.
Dereference: gets the data at the address which is being pointed to by a pointer. Defined only on pointer types. 4 marks

d) The value of myVariable 1 mark

e) The address of the variable. 1 mark

Question 2: (6 marks)

a) 0x2000 0124 + (42*4) = 0x2000 01CC 2 marks

b) 122 1 mark

c) The sizeof operator tells us how much memory has been allocated for a variable. When sizeof is called on an array type, it tells us the total amount of memory allocated to the array. Then sizeof(array[0]) is called it tells us the total memory allocated for one element.
(total memory) / (memory per element) = (number of elements) 2 mark

d) &(foo[42]) brackets are recommended but not necessary 1 mark

Question 3: (6 marks)

a) When a DIP switch is toggled, it may bounce for a couple of milliseconds, producing multiple edges which would cause the LCD to latch in the applied command or character multiple times.
The components act as a low pass filter, ensuring that the line can only go high or low at a slow speed, slow enough that the fast bouncing signal will not be allowed through to the LCD. 3 marks

b) The first, 001111xx sets 8 bit interface, 2 line display and 5x10 dot format.
The second, 00001111 sets the display on and enables underlined and blinking cursor. 3 marks