5 Programming in C and C++ (djg11+am21)

(*a*) 14-bit words are used to represent a certain set of natural numbers including zero. The least-significant five bits contain an unsigned binary-encoded mantissa value. The remaining nine bits represent an unsigned binary-encoded, bitwise left shift to be applied to the mantissa to obtain the represented value.

    (*i*) Give two functions, coded in C, that respectively convert an encoded value to its nearest 32-bit unsigned integer and to its nearest double-precision floating-point number. What problem(s) arise? [6 marks]

    (*ii*) A packed array of such 14-bit words is stored in memory. Packed means no memory bits are unused, so the stored words may cross byte boundaries. Write a C function to implement the update operation for a 14-bit word held in the array. Its three arguments are an `unsigned char *` pointer to the base of the packed array, an integer index and an integer holding the 14-bit word to be stored. You may assume unaligned loads and stores of 32-bit words is supported. [6 marks]

(*b*) All calls to `malloc()` in a user program in C are to be replaced with calls to `my_malloc`.

    (*i*) Provide an implementation of `void *my_malloc(size_t)` that invokes the system's underlying `malloc` but which adds 16 bytes of padding at the start and end of each allocated region which is initialised with a distinctive bit pattern. [3 marks]

    (*ii*) Provide a companion `my_free` function that checks for any changes to the starting pattern, reporting appropriately, or else continues to invoke the system's `free`. [3 marks]

    (*iii*) What might be the motivation for introducing `my_malloc`? [2 marks]