C Programming, Autumn 2013, Exercises for the Sixth Week

The picture contains a hidden message encoded in the least signicant bits of the pixel data.Write a function

1. Write a function

   ```
   uint8_t read_lsb(FILE *f);
   ```

   that reads one byte from a given FILE handle and returns the least signicant bit of that byte. Use the bitwise AND operator to extract the bit.

2. Write a function

   ```
   uint8_t extract_byte(FILE* f)
   ```

   that reads the next eight bytes from the FILE pointer, extracts the least significant bit from each and constructs a new byte out of the least significant bytes. The least-significant bit of the first byte becomes the least-significant bit of the new byte, the least-significant bit of the second btye become the second-least significant bit and so on. Return the constructed byte. Use the previous function to your advantage.

3. Write a function

   ```
   uint32_t read_uint32(FILE* f);
   ```

   that reads four bytes from a given FILE handle, interprets these bytes as a 32-bit unsigned little-endian integer and returns its value.

4. Write a function

   ```
   uint32_t read_pixel_offset(FILE* f);
   ```

   that seeks to file position 0xa in a given FILE handle and reads a 32-bit unsigned integer from that position. Test the function with the c-whoa.bmp file. The file is located under test/ folder. Note: The position 0xa (10) in a BMP image contains the byte offset of the pixel data of the image. In this BMP file, the pixel data starts from the byte offset 54.

5. Write the function

   ```
   void read_msg(FILE* f);
   ```

   that uncovers and prints the hidden message in the file given. Use the previous function to determine the location of pixel data in a given BMP file and move to that location in the file. From that position, extract the least significant bits into bytes by using the extract_byte function and print the bytes as characters to standard output. Keep on extracting and printing the bytes until you get a byte that has the value 0.

   Try your program with the c-whoa.bmp file – you should see the hidden message!