Hexadecimal Representations

1. Convert the following 32-bit binary patterns to octal and hexadecimal representation. (See note: *Bits and Bit Patterns*)

   (a) 01110010 11110101 00111101 00001001 _____

   (b) 10100010 11011111 11101001 00111000 _____

   (c) 01101110 10001111 10101101 01010010 _____

   (d) 01011001 11001110 00110111 10001101 _____

   (e) 10110001 11011001 11110101 01100100 _____

   (f) 01011101 10001110 00101111 10100011 _____

Character Representations

2. Convert the following bits to ASCII (8 bit) characters. (See note: *Representing Characters with Bit Patterns*)

   (a) 01001000 01100101 01101100 01101100 01101111 00100001 _____

   (b) 01000011 01001101 01010011 00110010 00110011 00110000_____

   (c) 01000110 01100001 01101100 01101100 00110001 00111000 _____

   (d) 01010010 01101111 00100001 00100001 01101001 01101110 01110011 _____

   (e) 01000011 00110000 01101100 01101100 01100101 01100111 01100101 _____

   (f) 01010100 00110100 01110010 01110011_____

3. Convert the following strings to a sequence of bytes. (Recall that in C, strings are really just an array of characters, terminated by the null character.) Give your answer in hex notation. (See note: *Representing Characters with Bit Patterns*)

   (a) `"Ab12"` _____ (d) `"U(9)"` _____

   (b) `"4& hH"` _____ (e) `"$_mY"` _____

   (c) `"y%Z6"` _____ (f) `"@Gm8"` _____

Other Base Systems

4. Convert the following quantities to their base-10 (decimal) representations. (See note: *Number Representation*)

   (a) 102_3 _____ (d) 515_6 _____

   (b) 41_5 _____ (e) 111_4 _____

   (c) 62_8 _____ (f) 614_7 _____

5. Convert the following decimal values to the specified base system. (See note: *Binary and Hex Integer Representation*)

   (a) 342 to base 3 _____ (d) 5023 to base 6 _____

   (b) 189 to base 4 _____ (e) 4782 to base 7 _____

   (c) 1229 to base 5 _____ (f) 7612 to base 9 _____

Unsigned Integer Representation

6. Convert the following 8-bit patterns to positive decimal numbers. (See note: *Number Representation*)

   (a) 00001001 _____ (d) 10000101 _____

   (b) 00111000 _____ (e) 01100100 _____

   (c) 01010010 _____ (f) 11001111 _____

7. Convert the following positive decimal numbers to 8-bit binary numbers. (See note: *Binary and Hex Integer Representation*)

   (a) 18 _____ (d) 108 _____

   (b) 25 _____ (e) 243 _____

   (c) 63 _____ (f) 175 _____

Signed Integer Representation

8. Convert the following negative decimal numbers to 8-bit 2's complement binary numbers. (See note: *Binary Addition and Two's Complement*)

   (a) -29 _____ (d) -15 _____

   (b) -86 _____ (e) -105 _____

   (c) -63 _____ (f) -71 _____

9. Convert the following 8-bit patterns to signed decimal numbers. Assume these patterns use 2's complement representation. (See note: *Binary Addition and Two's Complement*)

   (a) 10001011 _____ (d) 10011101 _____

   (b) 10111000 _____ (e) 10001111 _____

   (c) 10101011 _____ (f) 10010011 _____

10. Perform the following 2's complement additions. Give the end result (in binary format) and state whether or not an overflow occurs. (See note: *Binary Addition and Two's Complement*)

    (a)
    ```
     0111 0011
     1101 0010
    -----------
    ```

    (b)
    ```
     1111 0111
     1001 1010
    -----------
    ```

    (c)
    ```
     1111 0111
     0101 1010
    -----------
    ```

    (d)
    ```
     1101 0111
     1001 1010
    -----------
    ```

    (e)
    ```
     1101 0111
     1001 1010
    -----------
    ```

    (f)
    ```
     0101 0111
     0011 1010
    -----------
    ```

Floating Point Representation

11. Convert the following decimal values to IEEE-754 single precision (32-bit) floating point format. (See note: *Floating Point Representation*)

    (a) -2.875 _____

(b) 3.375 _____

(c) 6.75 _____

(d) 27.0 _____

(e) -12.5 _____

(f) -14.875_____

12. Convert the following patterns to a decimal value, assuming they are IEEE-754 single precision (32-bit) floating point format. (See note: *Floating Point Representation*)

(a) 01000001 10101100 00000000 00000000 _____

(b) 01000000 11101100 00000000 00000000 _____

(c) 01000001 01111100 00000000 00000000 _____

(d) 01000001 01100000 00000000 00000000 _____

(e) 11000001 11111100 00000000 00000000 _____

(f) 11000001 11101110 00000000 00000000 _____