ACTIVITY 14

1. Suppose you have an 8-bit unsigned integer *n*. Write expressions for each of the following, using &, |, ⊕, and ¬ to denote bitwise AND, OR, XOR, and NOT, respectively.

   a. *n* with bits 2 and 6 set

   b. *n* with bits 3 through 5 cleared

   c. *n* with the MSB flipped

   d. *n* with all of the bits flipped

   e. *n* with only the lowest 4 bits retained

   f. The value 1 if *n* is odd and 0 if *n* is even

   g. The value 0 if *n* is odd and 1 if *n* is even

2. What are the following (unsigned) numbers in decimal?

		Binary	Decimal
$1 \ll 0$	=	00000001_2	=
$1 \ll 1$	=	00000010_2	=
$1 \ll 2$	=	00000100_2	=
$1 \ll 3$	=	00001000_2	=
$1 \ll 4$	=	00010000_2	=
$1 \ll 5$	=	00100000_2	=

   So, in general, $1 \ll n$ is equal to what mathematical expression?

3. Suppose you have an 32-bit signed integer *n*. Write expressions for each of the following, using «, $\gg^u$, and $\gg^s$ to denote left shift, logical right shift, and arithmetic right shift, respectively.

   a. The value 1 if *n* is negative and 0 otherwise

   b. The value –1 if *n* is negative and 0 otherwise

4. Suppose you want to "pack" two four-bit unsigned integers into an 8-bit integer. For example, you could store the integers $0010_2 = 2$ and $1001_2 = 9$ by storing the 8-bit integer $00101001_2 = 41$: the first integer (0010_2) is stored in the upper 4 bits and the second integer (1001_2) in the lower 4 bits.

   a. Suppose AL contains the 8-bit value. Write a sequence of assembly language instructions that will "unpack" this value, placing the integer value from the upper 4 bits into AH and the lower 4 bits into AL. (There are several ways to do this; pick one.)

   b. Now, suppose AH and AL contain unsigned integers in the range $[0, 2^4 - 1]$. Write a sequence of instructions that will "pack" them into an 8-bit integer value, storing it in AL.