Lab Assignment #02: Print Binary

Goal: A stronger understanding of unsigned and signed (2's complement) binary representation.

Objective: To implement a C program that accesses the internal binary representation of an integer.

Assignment: (Read the extra credit part before beginning!)

Write a C program that does the following:

1. Read an integer value from the keyboard into an integer variable using scanf.

   Note: Use an unsigned integer variable. This will let you manipulate the bits more easily. Even though the variable is unsigned, use the "%d" format specifier to read the value so that the decimal input value can be signed.

2. Extract the least-significant eight bits of the integer into an array of binary digits. Each value stored in the binary array should only be 0 or 1.

   Note: Use the remainder of division by 2 (n % 2) to get the least-significant bit and store it in the first element of the array (subscript position 0). Divide the integer by 2 and repeat until you have all eight bits.

3. Use a loop to print the 8-bits of the binary number so that the most-significant bit (bit 7) is in the left-most position.

When you run your program, the screen should look something like:

```
Enter a decimal integer: 123
The 8 binary bits are: 01111011
```

Verify that your program operates correctly for several unsigned numbers in the range 0-255 and for several signed numbers in the range -128 to +127, including both positive and negative values as well as the minimum and maximum values of each range. Demonstrate your program to the teaching assistant.

5% Extra Credit: (This makes it easier to reuse some of this code in later labs!)

Organize your program into three functions:

1. A function with two parameters – one that is a single integer variable, and a second that is an array of eight binary digits. This function extracts the bits from the first variable and fills the elements of the array. This function should NOT call printf or scanf.

   Function prototype: void Dec2Bin(int decimal, int binary[8]) ;

2. A function with one parameter – an array of eight binary digits (0's or 1's). This function should display the eight bits (and nothing else) so that the most-significant bit is in the left-most position.

   Function prototype: void PrintBin(int binary[8]) ;

3. The main program, which should call the first two functions. The main program should prompt the user for input, get the decimal integer, call the first function to convert the integer to an array of binary digits, display an output label, and then call the second function to display the result.