Lab Assignment #05: Binary Addition and Subtraction

Goal: A stronger understanding of binary addition and subtraction.

Objective: To implement a C program that performs addition and subtraction in binary.

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

Write a C program that does the following:

1. Read an ASCII string of eight 0's and 1's from the keyboard into an array of chars using the C library function gets. Convert the string into an array of exactly 8 binary digits (integers 0 and 1), so that the least-significant digit is in the first element (subscript 0) of the binary array.

   Note: Use code from your solution to Lab 3.

2. Repeat step 1 to fill a second array of 8 binary digits.
3. Compute the contents of a third array of 8 binary digits, corresponding to the sum of the first two binary arrays. Also compute the carry-out.
4. Display the resulting binary sum and carry-out.

   Note: Use code from your solution to Lab 2.

5. Recompute the contents of the third array of 8 binary digits, corresponding to the difference of the first less the second binary array. Also compute the borrow-out.
6. Display the resulting binary difference and borrow-out.

   Note: Use code from your solution to Lab 2.

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

```
Enter the 1st binary number: 11110000
Enter the 2nd binary number: 00010001
The 1st plus the 2nd is: 00000001
The carry out is: 1
The 1st less the 2nd is: 11011111
The borrow out is: 0
```

Verify that your program operates correctly for binary input strings in the range 00000000 to 11111111, including several combinations of positive, negative and end-of-range values. 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 five functions:

1. Function Asc2Bin that was described in Lab 3.
2. A new function with three parameters – each is an array of eight binary digits (integer 0's or 1's). This function computes the contents of the third array as the sum of the first two and returns the carry out as the value of the function.

   Note: The first element of each binary array (subscript position 0) is the least-significant bit. This function should NOT call printf or scanf.

   Function prototype: int AddBinary(int first[8], int second[8], int result[8]) ;

3. A new function with three parameters – each is an array of eight binary digits (integer 0's or 1's). This function computes the contents of the third array as the difference of the first two and returns the borrow out as the value of the function.

   Note: The first element of each binary array (subscript position 0) is the least-significant bit. This function should NOT call printf or scanf.

   Function prototype: int SubBinary(int first[8], int second[8], int result[8]) ;

4. Function PrintBin that was described in Lab 2.

5. The main program, which should call the other functions to compute the results as displayed in the sample output shown above.