Cse2421 Programming Assignment 2 September 14, 2017
Due: Wednesday, September 20, 2017

1. Make the function BinaryAdd that performs addition of two 32-bit integers at bit level. The function ***must*** use algorithm based on 1-bit adder truth table on slide D25. If you use any other algorithm, your code will not be graded and you will be given 0 points for this assignment. Function declaration:

   int BinaryAdd(int in1, int in2);

   where in1 and in2 are two operands to add and the function returns the sum.
   Allowed operations: assignments, all bit-level operations (&, |, ^, ~), all logical operations (&&, ||, !), left (<<) and right(>>) shifts, equality (==) and inequality (!=). Also, you can have only this loop (i.e. for statement that uses addition): int i; for (i=0; i<32; i++) { ….. }, since your algorithm has to have 32 iterations. Global variables and arrays are not allowed.

2. Your main function gets as input 2 integers (use scanf with %x), prints them (as signed, unsigned (use %u) and hex), then it calls BinaryAdd, then it prints results from BinaryAdd (as signed, unsigned and hex) and indicates if there is signed or/and unsigned overflow. You may assume that two integers on input are provided without any illegal character. In additions to allowed operations in 1. above, you may use less than operation (<) only once in the function main.

Here are examples how input/output of this program could look like:

```
~/Cse2421/Lab2> BiAdd
Give 2 integers in hex to add:56  78
First = 0x00000056   As unsigned= 86   As signed= +86
Second= 0x00000078   As unsigned= 120   As signed= +120
Sum   = 0x000000CE   As unsigned= 206   As signed= +206

~Cse2421/Lab2> BiAdd
Give 2 integers in hex to add:ffffffff   2
First = 0xFFFFFFFF   As unsigned= 4294967295   As signed= -1
Second= 0x00000002   As unsigned= 2   As signed= +2
Sum   = 0x00000001   As unsigned= 1   As signed= +1
Unsigned overflow

~/Cse2421/Lab2> BiAdd
Give 2 integers in hex to add:7fffffff   8
First = 0x7FFFFFFF   As unsigned= 2147483647   As signed= +2147483647
Second= 0x00000008   As unsigned= 8   As signed= +8
Sum   = 0x80000007   As unsigned= 2147483655   As signed= -2147483641
Signed overflow
```

Submission: A **hard** copy AND electronic copy of your source code.
For electronic submission use: submit c2421ac lab2 your_source_code