Programming Assignment 3: Binary Adder-Subtractor

Write a C program to simulate the combinational logic design of a 32-bit binary adder-subtractor. A 32-bit binary adder-subtractor is a logic circuit of 32 one-bit full adders. When adding/subtracting two 32-bit integers, S=X+Y or S=X-Y, an additional bit M is used to indicate addition or subtraction operation. If the operator is "+", M is set to 0; if the operator is "-", M is set to 1. The initial carry-in bit c_in is set to M. A full adder takes x_i, y'_i , and c_{in} as input and produces s_i and c_{out}, where x_i and s_i are the i-th bit of X and S, respectively, y'_i is $y_i \square M$. The carry-out bit c_{out} of the i-th full adder becomes c_{in} of the (i+1)-st full adder. The logic formula of a full adder is defined as below:

$$s_i = (x_i \oplus y'_i) \oplus c_{in}$$
$$c_{out} = (x_i \wedge y'_i) \vee (c_{in} \wedge (x_i \oplus y'_i))$$

Refer to Digital System Design Lecture 12, Combinational Logic Design Binary Adder-Subtractor (binary_adder_subtractor.pdf) for more details of the logic design of binary adder-subtractor. The program will repeatedly input two 32-bit integers X and Y, and use a binary adder-subtractor to compute S=X+Y or =X-Y, until both X and Y are 0's. **Do not** use addition/subtraction operation in C programming language. The output will print X, Y, and S in both decimal and binary format. Also, print a message to confirm that the binary adder-subtractor has the same result as the addition/subtraction operation of C programming language. If the addition/subtraction results in the overflow situation, print an overflow message. Repeat the program until both X and Y are 0. (Hint: Use "scanf("%d %c %d", &X, &op, &Y);" to enter expression "X + Y" or "X - Y".)

Write comments in your program solution. Also, write a report to explain how you develop your assignment solution. Homework assignment 3 is due by **11:59 pm, Monday, October 24**. Use **assgn3_DXXXXXXX.c** for your source code file and **assgn3_DXXXXXXX.pdf** for your report. where DXXXXXXX is your student ID. Submit the source code and the report to **iLearn2**.

Example of program execution (next page):

```
命令提示字元

D:\>binary_adder_subtractor
Enter "X + Y" or "X - Y" (X, Y: -2,147,483,648 to 2,147,483,647): 28 + 45
X = 28           Binary value: 0000 0000 0000 0000 0000 0000 0001 1100
Y = 45           Binary value: 0000 0000 0000 0000 0000 0000 0010 1101
S = 73           Binary value: 0000 0000 0000 0000 0000 0000 0100 1001
Correct! Adder-subtractor operation test: 28 + 45 = 73
--------------------------------------------------
Enter "X + Y" or "X - Y" (X, Y: -2,147,483,648 to 2,147,483,647): 28 - 45
X = 28           Binary value: 0000 0000 0000 0000 0000 0000 0001 1100
Y = 45           Binary value: 0000 0000 0000 0000 0000 0000 0010 1101
S = -17          Binary value: 1111 1111 1111 1111 1111 1111 1110 1111
Correct! Adder-subtractor operation test: 28 - 45 = -17
--------------------------------------------------
Enter "X + Y" or "X - Y" (X, Y: -2,147,483,648 to 2,147,483,647): 28 + -45
X = 28           Binary value: 0000 0000 0000 0000 0000 0000 0001 1100
Y = -45          Binary value: 1111 1111 1111 1111 1111 1111 1101 0011
S = -17          Binary value: 1111 1111 1111 1111 1111 1111 1110 1111
Correct! Adder-subtractor operation test: 28 + -45 = -17
--------------------------------------------------
Enter "X + Y" or "X - Y" (X, Y: -2,147,483,648 to 2,147,483,647): -28 + 45
X = -28          Binary value: 1111 1111 1111 1111 1111 1111 1110 0100
Y = 45           Binary value: 0000 0000 0000 0000 0000 0000 0010 1101
S = 17           Binary value: 0000 0000 0000 0000 0000 0000 0001 0001
Correct! Adder-subtractor operation test: -28 + 45 = 17
--------------------------------------------------
Enter "X + Y" or "X - Y" (X, Y: -2,147,483,648 to 2,147,483,647): -28 - 45
X = -28          Binary value: 1111 1111 1111 1111 1111 1111 1110 0100
Y = 45           Binary value: 0000 0000 0000 0000 0000 0000 0010 1101
S = -73          Binary value: 1111 1111 1111 1111 1111 1111 1011 0111
Correct! Adder-subtractor operation test: -28 - 45 = -73
--------------------------------------------------
Enter "X + Y" or "X - Y" (X, Y: -2,147,483,648 to 2,147,483,647): -28 - -45
X = -28          Binary value: 1111 1111 1111 1111 1111 1111 1110 0100
Y = -45          Binary value: 1111 1111 1111 1111 1111 1111 1101 0011
S = 17           Binary value: 0000 0000 0000 0000 0000 0000 0001 0001
Correct! Adder-subtractor operation test: -28 - -45 = 17
--------------------------------------------------
Enter "X + Y" or "X - Y" (X, Y: -2,147,483,648 to 2,147,483,647): 45 - 28
X = 45           Binary value: 0000 0000 0000 0000 0000 0000 0010 1101
Y = 28           Binary value: 0000 0000 0000 0000 0000 0000 0001 1100
S = 17           Binary value: 0000 0000 0000 0000 0000 0000 0001 0001
Correct! Adder-subtractor operation test: 45 - 28 = 17
--------------------------------------------------
Enter "X + Y" or "X - Y" (X, Y: -2,147,483,648 to 2,147,483,647): 45 - -28
X = 45           Binary value: 0000 0000 0000 0000 0000 0000 0010 1101
Y = -28          Binary value: 1111 1111 1111 1111 1111 1111 1110 0100
S = 73           Binary value: 0000 0000 0000 0000 0000 0000 0100 1001
Correct! Adder-subtractor operation test: 45 - -28 = 73
--------------------------------------------------
Enter "X + Y" or "X - Y" (X, Y: -2,147,483,648 to 2,147,483,647): 1000000000 + 1000000000
X = 1000000000   Binary value: 0011 1011 1001 1010 1100 1010 0000 0000
Y = 1000000000   Binary value: 0011 1011 1001 1010 1100 1010 0000 0000
S = 2000000000   Binary value: 0111 0111 0011 0101 1001 0100 0000 0000
Correct! Adder-subtractor operation test: 1000000000 + 1000000000 = 2000000000
--------------------------------------------------
Enter "X + Y" or "X - Y" (X, Y: -2,147,483,648 to 2,147,483,647): 1000000000 - 2000000000
X = 1000000000   Binary value: 0011 1011 1001 1010 1100 1010 0000 0000
Y = 2000000000   Binary value: 0111 0111 0011 0101 1001 0100 0000 0000
S = -1000000000  Binary value: 1100 0100 0110 0101 0011 0110 0000 0000
Correct! Adder-subtractor operation test: 1000000000 - 2000000000 = -1000000000
--------------------------------------------------
Enter "X + Y" or "X - Y" (X, Y: -2,147,483,648 to 2,147,483,647): 1000000000 + 2000000000
X = 1000000000   Binary value: 0011 1011 1001 1010 1100 1010 0000 0000
Y = 2000000000   Binary value: 0111 0111 0011 0101 1001 0100 0000 0000
S = -1294967296  Binary value: 1011 0010 1101 0000 0101 1110 0000 0000
Correct! Adder-subtractor operation test: 1000000000 + 2000000000 = -1294967296
**** The addition-subtraction operation is overflow.
--------------------------------------------------
Enter "X + Y" or "X - Y" (X, Y: -2,147,483,648 to 2,147,483,647): -1000000000 - 2000000000
X = -1000000000  Binary value: 1100 0100 0110 0101 0011 0110 0000 0000
Y = 2000000000   Binary value: 0111 0111 0011 0101 1001 0100 0000 0000
S = 1294967296   Binary value: 0100 1101 0010 1111 1010 0010 0000 0000
Correct! Adder-subtractor operation test: -1000000000 - 2000000000 = 1294967296
**** The addition-subtraction operation is overflow.
--------------------------------------------------
Enter "X + Y" or "X - Y" (X, Y: -2,147,483,648 to 2,147,483,647): 0 + 0
微軟注音 半 :
```