2003-2004 **American Computer Science League** **Problem #3**

All-Star Contest

ACSL COMPLEMENT

PROBLEM: Computers are designed with an add function. No separate subtraction function exists. Subtraction is handled using a process called 2's complement arithmetic. The computer converts the subtrahend to its 2's complement and adds it to the minuend.

In the problem "Subtract 185 (the subtrahend) from 230 (the minuend)", 230 – 185, the computer first changes both numbers to their binary representation using the number of bits required by the larger number.

$230_{10} = 11100110_2$ $185_{10} = 10111001_2$

It then adds a left most bit to each number to denote its sign. 0 is used to denote positive and 1 is used to denote negative. The representations are now as follows:

$230_{10} = 011100110_2$ $185_{10} = 010111001_2$

The 2's complement of a binary number is found using the following algorithm:

1. Change the 1's to 0's and the 0's to 1's.
2. Add 1 to the result

The 2's complement of the subtrahend (185) is computed as follows:

$101000110_2 + 1_2 = 101000111_2 = -185_{10}$

Adding the binary representation of 230_{10} and the 2's complement of 185_{10} gives:

$$\begin{array}{r} 011100110_2 \\ 101000111_2 \\ \hline 1000101101_2 \end{array}$$

Note that the addition of the sign bits produced an extra bit caused by the "carry" rule. This extra bit is dropped. The result is now 000101101_2 which converts to $+ 45_{10.}$ The left-most bit produces the + sign.

In a somewhat similar manner we can compute $185_{10} – 230_{10}$. This time we need to find the 2's complement of 230, and add it to binary representation of 185.

$$\begin{array}{rr} 230 = & 100011010_2 \\ 185 = & 010111001_2 \\ \hline & 111010011_2 \end{array}$$

Note that a "carry" bit was not produced. The left-most 1 denotes that the number is negative. When the sum is negative, delete the sign bit and take the 2's complement of the remaining bits to find its numerical value. This gives 00101101_2, which is equivalent to 45_{10}.

INPUT: There will be 2 input lines. Each line will contain two base 10 numbers in minuend – subtrahend order.

OUTPUT: For each input print: 1) the base 2 representation of the minuend, 2) the base 2 representation of the subtrahend, 3) the 2's complement of the subtrahend, 4) the 2's complement of the minuend and 5) the sum of the minuend and the 2's complement of the subtrahend if the sum is positive or the 2's complement of that sum if it is negative. Note that in SAMPLE OUTPUTS 5 and 10, the exact number of leading 0's that are produced by applying the above rules must be shown. Note also that OUTPUTS 1 and 2 and 5 and 6 must be of the same length.

SAMPLE INPUT

1. 230, 185
2. 185, 230

SAMPLE OUTPUT

1. 11100110
2. 10111001
3. 101000111
4. 100011010
5. 000101101
6. 10111001
7. 11100110
8. 100011010
9. 101000111
10. 00101101