Lab 5: Bit Manipulation

Requirements:

1. This assignment as well as other assignments in this class must be finished on Windows operating system.
2. Zip your program(s) and submit on Canvas
3. Due 11:59pm Nov. 5, 2013

Assignments:

Part 1

Implement the procedure "SwapOddEvenBits" in the given incomplete program. The logic for this procedure is given in the Java program. This procedure is supposed to, given a DWORD, exchange the bits in every consecutive even and odd positions. See the following examples:

- Before: 0110 1110 0000 1001 0011 1000 0111 0010
- After: 1001 1101 0000 0110 0011 0100 1011 0001

- Before: 1010 0001 0000 1100 0100 0100 1110 1011
- After: 0101 0010 0000 1100 1000 1000 1101 0111

Basically, the idea is to isolate every pair of the bits in the DWORD integer, using *shl(shr)* and *and* instructions. If the two bits are either "00" or "11", there is no need to swap them. Otherwise, xor the bits and "11" to flip both of them. I will explain the idea in detail during the class.

Part 2

Implement the procedures "CountDistance" and "CountOnes". They can be used to find how many pairs of bits in two byte vectors are different. Basically, the idea is to xor two bit sequences. In the result bit sequence, a one means a pair of bits being different. Then it needs to find how many bits in the xor result are one. See the following example:

- Byte 1: 1001 0110
- Byte 2: 0111 1011
- Xor: 1110 1101

To find how many bits are one in a bit sequence, the most straightforward way is to examine every bit in the sequence, and use cmp and jumps to find how ones are there. Obviously, the performance of this algorithm has nothing to do with how many ones are there. A better algorithm will be introduced during the class. The algorithm is given in the Java program.

I will explain the algorithms in detail during the class.

Grading Policies:

Part 1	50%
Part 2	50%