CMPE-180-93 Homework 4

Using the class you implemented in Homework 3, extend the shift register class to support the plus (‘+’) and minus (‘-‘) operator. The semantic of those operators would be that of binary addition and subtraction. You most modify your command line argument parser to support passing two registers as follows: (ARGUMENTS MUST BE SUPPORTED IN ANY ORDER.)

by default your register must inject 0 values unless told otherwise via command line arguments.

Here are some of the parameters to support as your command line arguments:

-i “010101”: Initial values stored in your register1. . This option must accept a positive or negative integer and covert it to binary.

-s 6: Specifies the number of bits in your shift register1.

-r 2: shift the register1 right by two positions.

-l 3: shift left register1 by 3 positions.

-I “010101”: Initial values stored in your register2. This option must accept a positive or negative integer and covert it to binary.

-S 6: Specifies the number of bits in your shift register2.

-R 2: shift the register2 right by two positions.

-L 3: shift left register2 by 3 positions.

-o +/-: operator to be used among the registers if any

-v 1: value to inject in vacated bits if other than default.

-d: if this option is given the resulting registers are followed by their representation in Decimal as follows: 0111(7)

-h: as above but in hex:1111(0xF)

-p: prints the value of bits in your register after performing all the operations. Bits must be printed as a non-spaced string of 0 or 1s and terminated by a new line. One line for register1, one line for register2 and the last line contains the result in binary. Implement the << operator in your register class so you can output using cout << register

You must turn in ONLY ONE UNCOMPRESSED TEXT FILE (NO INCLUDES, NO BINARIES): the c++ file named shiftregister.cpp

Multiple operations may be specified and they must be performed in the order specified in the command line argument list.

Any violation of the specs is likely to result on a zero grade especially if they prevent your program from compiling or running without core dumps.

VERY IMPORTANT: your program must return 0 on success and -1 on failure, not abiding to this will result in zero credit in the full assignment.