Write a C/C++ program to imitate algorithm presented in the MSDAP paper. (Please use only shift and addition operation for computation)

(1.1) Use the provided coefficients and inputs.

(1.2) All the input samples are signed 16-bit hex number, two's complement, fixed-point, MSB. Leftmost bit is sign bit.

(1.3) keep the whole 40 bits as the final output data and convert it into hexadecimal number.

(1.4) Input and coefficients are read in a serial manner. So the output is also latched out at a serial manner.

(1.5) The filter order is N=255.

(1.6) Total number of input sample is n=1000.

(1.7) Assume x(-255)=x(-254)=⋯=x(-2)=x(-1)=x(0)=0.

(1.8) Save your output file as output.txt.

(1.9) Follow the procedure in the example for computation.

Please submit your Homework #3 which contains:

(a) Your C/C++ source code.

(b) Your output.txt.


**Computation transformation**

Example:

Assume filter order $N=3$, POT digit limit to $2^4$ (in this HW, it can reach $2^{16}$)

$$y(n) = \sum_{k=0}^{3} h(k)x(n-k)$$

$$= h(0)x(n-0) + h(1)x(n-1) + h(2)x(n-2) + h(3)x(n-3) \quad (n=1:1000)$$

$$h(0) = 2^{-1} - 2^{-3}$$

Assume

$$h(1) = 2^{-3} + 2^{-4}$$

$$h(2) = 2^{-1} + 2^{-2} - 2^{-4}$$

$$h(3) = -2^{-3}$$

$$y(n) = 2^{-1}[x(n-0) + x(n-2)] + 2^{-2}[x(n-2)] + 2^{-3}[-x(n-0) + x(n-1) - x(n-3)] + 2^{-4}[x(n-1) - x(n-2)]$$

Let

$$u_4 = x(n-0) + x(n-2)$$

$$u_3 = x(n-2)$$

$$u_2 = -x(n-0) + x(n-1) - x(n-3)$$

$$u_1 = x(n-1) - x(n-2)$$

$$y(n) = 2^{-1}u_4 + 2^{-2}u_3 + 2^{-3}u_2 + 2^{-4}u_1$$

$$= 2^{-1}(u_4 + 2^{-1}(u_3 + 2^{-1}(u_2 + 2^{-1}u_1)))$$

**Coefficient and input format**

**The format of input data:**

C48B=

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSB</td>
<td>16 bits</td>
<td>LSB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

C48B: 1100 0100 1000 1011
ext. to 24-bit: 1111 1111 1100 0100 1000 1011 (left 8 bits are same to the sign bit, give enough room for carry in bits generated during the computation)

two’s complement: 0000 0000 0011 1011 0111 0101
C48B*2⁻¹: 1111 1111 1110 0010 0100 0101 1000 0000 0000 0000
C48B*2⁻²: 1111 1111 1111 0001 0010 0010 1100 0000 0000 0000
...
C48B*2⁻¹₆: 1111 1111 1111 1111 1111 1111 1100 0100 1000 1011
(right 16 bits are for the shift operation)

The format of coefficient(h(k)) data:

\[
\begin{array}{|c|c|c|c|}
\hline
\text{Sign bits} & D & \text{POT digits} \\
\hline
1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
\hline
9 & 1 & 4 & 8 \\
\hline
\end{array}
\]

\[h(1)=-2^{-1}+2^{-4}+2^{-8}+2^{-10}+2^{-13}\] (ignore the sign bits (highlighted in yellow) that are not used when POT digit is 0)