In this assignment you will write an OpenSPL application that receives eight input streams (a, b, c, d, e, f, g, h) and two scalars (i, j) and generates two output streams (s1, s2).

All streams and scalar variables will be 32-bit integers.

Calculate the following streaming output values:

\[
    s_1 = (a + b + c) * i + (d + e + f + g + h) * j; \\
    s_2 = (a + i) * (b + i) * (c + i) + (d * e * f * g * h) + j; 
\]  

(1)  

(2)  

Note that the above math is to be interpreted in the context of an OpenSPL expression. In a C program the expression for s2 might be expressed as:

```c
for(int k = 0; k < size; ++k)
    s2[k]=(a[k]+i)*(b[k]+i)*(c[k]+i)+(d[k]*e[k]*f[k]*g[k]*h[k])+j;
```

Create 96-element buffers for your test streams and fill them with suitable test data. Note that using large values for test data could result in overflow.

1 Compiling and Testing Your Program

Build and test your program using the VM as discussed in class.

2 How to Hand in Your Program

Hand in your program using the SVN repo as discussed in class and in the appendix of the lecture notes.

3 What Happens if....

After handing in your program, change the math for s2 to this:

\[
    s_2 = (a * b * c) / i + (d * e * f * g * h) / j; 
\]  

(3)  

... then rebuild and run it. What happens? Why?