The display thread module shown below prints cell states of successive generations of the GameOfLifeArray stream module.

class display: public threadModule
{
    inputStream<bool> inStrm[nROWS][nCOLS];    // Input streams

    void code() { // Thread-domain code
        int g, r, c;
        bool cellState;

        for (g = 0; g < nGENERATIONS; ++g) // Print generations
        {
            printf( "\nGENERATION %d", g);
            for (r = 0; r < nROWS; ++r)
            {
                printf( "\n" );
                for (c = 0; c < nCOLS; ++c)
                {
                    inStrm[r][c] >> cellState; // Get cell state
                    printf( "%s", ((cellState == 0) ? " -" :
                                      " X") );
                }
            }
            printf( "\n" );
        }
        printf( "\nDone\n
" );
        endTopology(0);
    }
};

Some observations about display:

a. The thread-domain code for display gets cell states from inStrm in *row-major order*\(^1\), and prints to the console in the same order.

b. For each generation, display prints a grid of ‘-’s and ‘X’ s with ‘-’ representing a *dead* cell and ‘X’ representing a *live* cell.

c. The statement
   
   endTopology(0);
   
calls the module member function
   
   void endTopology( int result )
   
The call causes:

\(^1\) [https://en.wikipedia.org/wiki/Row- and_column-major_order](https://en.wikipedia.org/wiki/Row- and_column-major_order)
• All thread modules in the TruStream topology to which display belongs to cease execution.

• The TruStream topology to which display belongs to be torn down.

• A completion signal to be sent to any wait() that is waiting for completion of the topology.