```
// Display.cpp - QUEUE DISPLAY DRIVER PROGRAM
//
// MODULE INDEX
// NAME                 CONTENTS
// Refresh              Refresh the display
// main                 Main line
//
// MAINTENANCE HISTORY
// DATE         PROGRAMMER AND DETAILS
// 05-10-15     JS      Original
//
//-----------------------------------------------------------------------------

#include <cstring>              // C-style string manipulation functions
#include <cctype>               // C-style character typing functions
#include <ctime>                // C-style system time functions
#include <iostream>             // C++ I/O stream declarations
#include <string>               // C++ string declarations
#include <vector>               // C++ vector declarations
using namespace std;    // Expand the standard namespace

//-----------------------------------------------------------------------------

// DISPLAY LINE DATA

struct Line_t {
        string          seqNo;          // Sequence number
        char            countNo;        // Counter number
};

//-----------------------------------------------------------------------------

// GLOBAL DATA

vector<Line_t>          lineVec;        // Display line vector

//-----------------------------------------------------------------------------

// REFRESH THE DISPLAY

void
Refresh (
        bool            flashOn)        // Flash on flag
{
        for (size_t i = 0; i < 4; i++) {
                if (i < lineVec.size()) {
                        if (i != 0 || flashOn)
                                cout << lineVec[i].seqNo << ' ';
                        else
                                cout << "     ";
                        cout << lineVec[i].countNo;
                }
                cout << '\n';
        }
}

//-----------------------------------------------------------------------------

// MAIN LINE

int
main ()
{
        Line_t          lineRec;        // Line record
```

```
        struct timespec sleepTime;      // Sleep time
        istream::int_type ch;           // Input character
        size_t          i;              // General purpose index

        Refresh (1);
        for (;;) {

                // Receive the next call

                lineRec.seqNo = "";
                for (i = 0; i < 4; i++) {
                        if ( ! isdigit(ch = cin.get())) goto bad_input;
                        lineRec.seqNo += static_cast<char>(ch);
                }
                if ( ! isdigit(ch = cin.get())) goto bad_input;
                lineRec.countNo = ch;
                if ((ch = cin.get()) != '\n') goto bad_input;

                // Process new and existing sequence numbers

                i = 0;
                while (i < lineVec.size() && lineVec[i].seqNo != lineRec.seqNo)
                        i ++ ;
                if (i < lineVec.size())
                        lineVec.erase (lineVec.begin() + i);
                else if (lineVec.size() >= 4)
                        lineVec.pop_back ();
                lineVec.insert (lineVec.begin(), lineRec);

                // Initiating the flashing refresh

                Refresh (1);
                for (i = 0; i < 3; i++) {
                        sleepTime.tv_nsec = 700000000;
                        sleepTime.tv_sec = 0;
                        nanosleep (&sleepTime, 0);
                        Refresh (0);
                        sleepTime.tv_nsec = 300000000;
                        nanosleep (&sleepTime, 0);
                        Refresh (1);
                }
                continue;

                // Process bad input

        bad_input:
                while (ch != '\n') ch = cin.get();
        }
        // NOTREACHED
        return 0;
}
```