```c
 * globals.c

#include "globals.h"
#include "stdio.h"

// The length of the sequence.
// The static keyword means that globals_sequenceLength can only be accessed
// by functions contained in this file.
static uint16_t globals_sequenceLength = 0;  // The length of the sequence.
static uint16_t globals_sequenceIterationLength = 0;
uint16_t seriesArray[GLOBALS_MAX_FLASH_SEQUENCE] = {0};

// This is the length of the sequence that you are currently working on,
// not the maximum length but the interim length as
// the user works through the pattern one color at a time.
void globals_setSequenceIterationLength(uint16_t length) {
    globals_sequenceIterationLength = length - 1;
}


// This is the length of the complete sequence at maximum length.
// You must copy the contents of the sequence[] array into the global variable that you maintain.
// Do not just grab the pointer as this will fail.
void globals_setSequence(const uint8_t sequence[], uint16_t length){
    for (uint16_t i = 0; i <= length; i++){
        seriesArray[i] = sequence[i];
    }
    globals_sequenceLength = length;
}

// This returns the value of the sequence at the index.
uint8_t globals_getSequenceValue(uint16_t index){
    return seriesArray[index];
}

// Retrieve the sequence length.
uint16_t globals_getSequenceLength(){
    return globals_sequenceLength;
}

// This is the length of the sequence that you are currently working on,
// not the maximum length but the interim length as
// the use works through the pattern one color at a time.
uint16_t globals_getSequenceIterationLength(){
    return globals_sequenceIterationLength;
}


// You will need to implement the other functions.


```