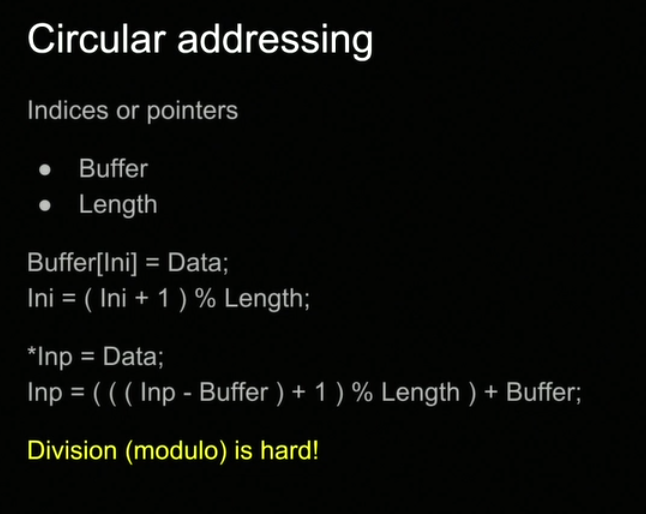
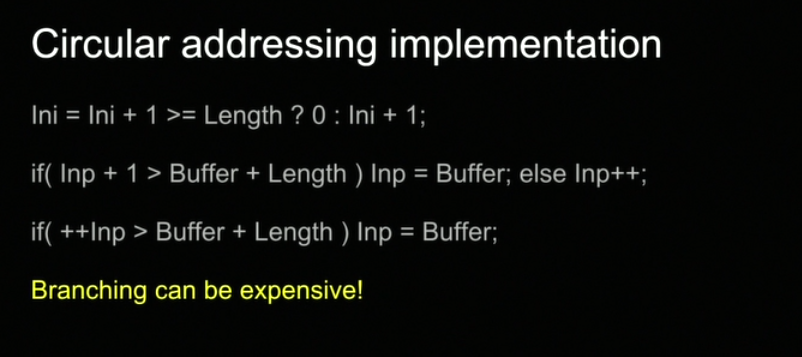
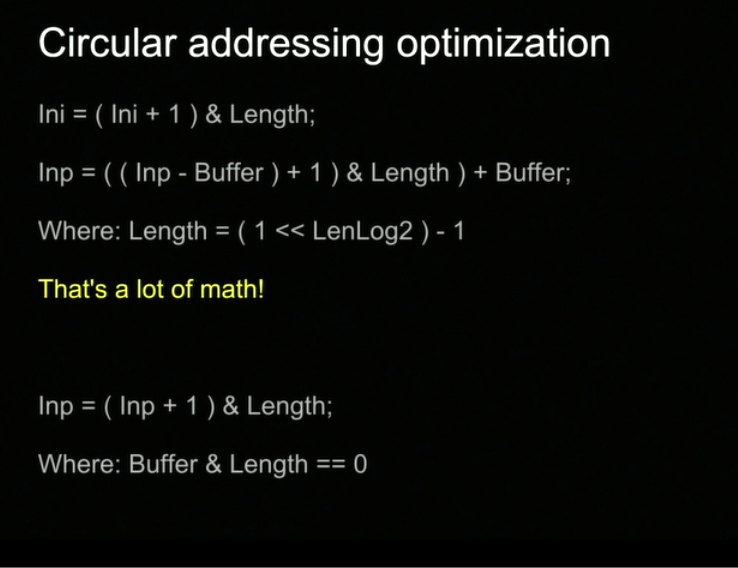
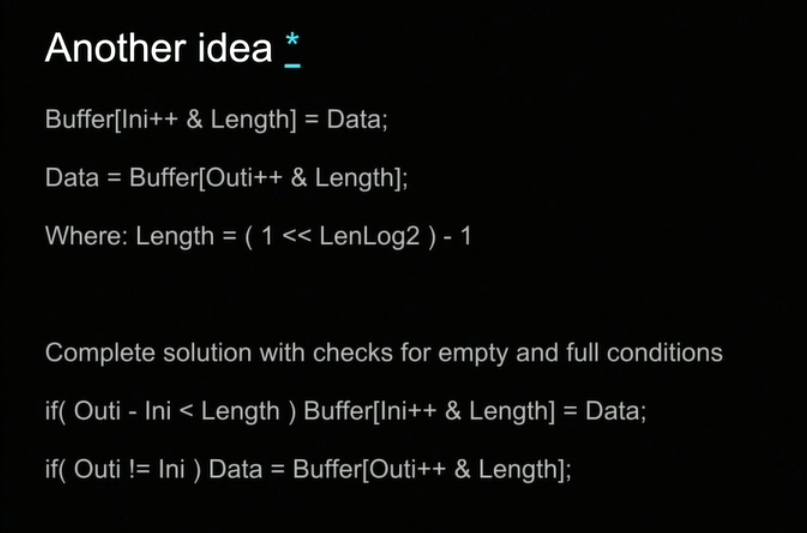
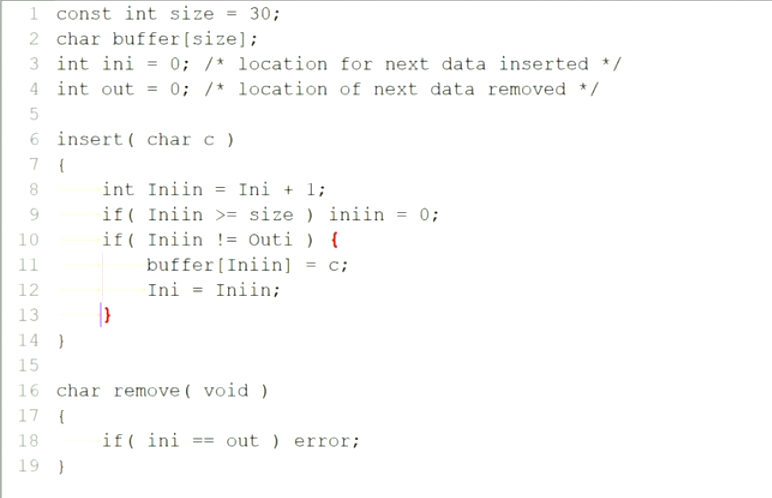
**Ring buffer implementation:**











Masking the MSB by[ Ini ++ & (length-1)], means if there is an 8 (length) location buffer (buffer[8]) then we must go from 0 to 7 then back to 0 (000 – 111, for length-1 = 7 (111) then the index will wrap 0 ~ 7 no matter how large Ini gets

**Homework#5:**

**Please review the**[**lecture notes**](https://canvas.colorado.edu/courses/24817/pages/circular-buffers)**[Preview the document](https://canvas.colorado.edu/courses/24817/pages/circular-buffers) and write code to implement the circular buffer of characters. At the bottom are three files that outline the circular buffer module we'll use as part of project 2. Function prototypes are on ring.h. You need to add a typedef for ring\_t to what is here. Add your implementation of the prototyped functions in ring.c.**

**ring\_test.c is for implementing unit tests. This will contain your main() function. The unit testing framework we'll use is [CUnit (Links to an external site.)Links to an external site.](http://cunit.sourceforge.net/index.html" \t "_blank). Since, due to campus closure, we haven't covered CUnit, I won't grade the quality of what you put into this file, you just get 5 points if you give it a crack. If you get CUnit working, you will be able to test your code and make it hum and probably get a better grade on the 10 points based on your ring.h and ring.c implementation.**

**I suggest you base ring\_test.c on the [CUinit example (Links to an external site.)Links to an external site.](http://cunit.sourceforge.net/example.html" \t "_blank). The hard part in any unit testing is coming up with your test cases. Test cases are written as assertions. The workhorse for CUint is the CU\_ASSERT macro which works like the standard**[**assert (Links to an external site.)Links to an external site.**](http://man7.org/linux/man-pages/man3/assert.3.html)**facility in C. The test cases you write should cover nominal operations: successful initialization, insert, remove. You should also minimally test expected error conditions: remove from empty buffer, insert to full buffer.**