CPTR-241 Inclass assignment and HW#4
Due Thursday, 18 January

Background

On pages 149 to 153 of our text a Queue class is described that uses an array to hold data and two indexes, one that indicates the next location where a new data item will be placed and the second the next location from which data will be read. Both indexes are only incremented and never decremented. Once the first index reaches the end of the array this routine is in trouble because all the space has been used. Note that a queue is a first-in first-out data structure.

Another way to create a queue is to use a "circular buffer". As long as the total number of items that need to be stored at any time is less than the array size, it will never run out of space because the array space will be used over and over again. Again there are two indices. One indicates the position in the array where the next item will be stored (putloc. which can be referred to as the tip of the buffer)) and other the position at which to read (getloc, the tail of the buffer). putloc is incremented as items are placed in the array until it gets to end of the array and then is set back to the start of the array (zero). Before incrementing putloc a check is made to see if this would cause writing over the oldest item that has not yet been read from the buffer. If so, the buffer is full.

Items are read from the location specified by getloc unless getloc equals putloc in which case the buffer is empty and there is nothing to read. After an item is read getloc is incremented. If incrementing moves getloc beyond the end of the array then it is set back to the start of the array.

To Do

Modify the Queue example program from the book to make it into a circular queue (I have placed the source code for it on the class web page to save you typing it in). The main part of the program can be used as is but the get and put methods need to change. Note that when the buffer is full there may be one unused spot in the array (thus when the smallQ object is used the contents of the buffer will be 3 characters rather than 4 as shown in the book's size array).

Extend the program or create a second version that allows entering characters from the keyboard, placing them into the queue, and when a $ character is entered displaying the contents of the queue.

At the top of all program code you always need to place this information as comment text:

- your name
- assignment number (i.e. HW#4)
- date

Submit your homework by uploading to D2L.

PLEASE

When doing this assignment please don't google for code that implements a circular buffer. Think through the logic and implement it. If needed, draw a picture of a small array, increment the index values as you place data in and take it out. The modifications to the put and get methods require changing and adding only a few statements.