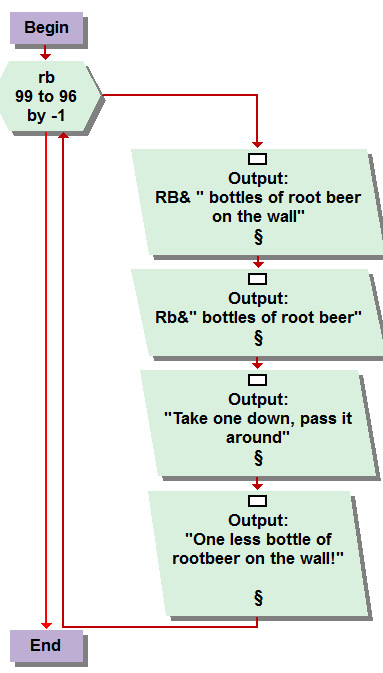
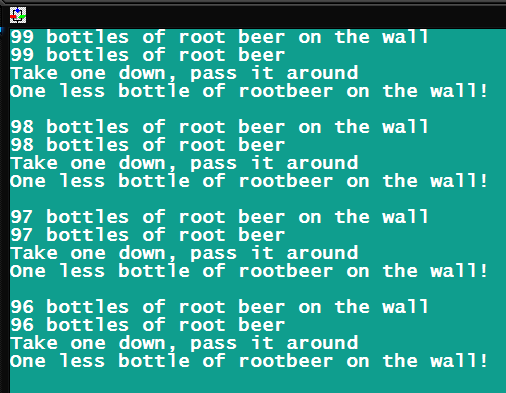
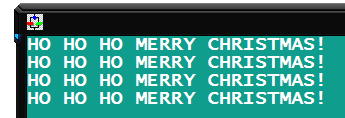
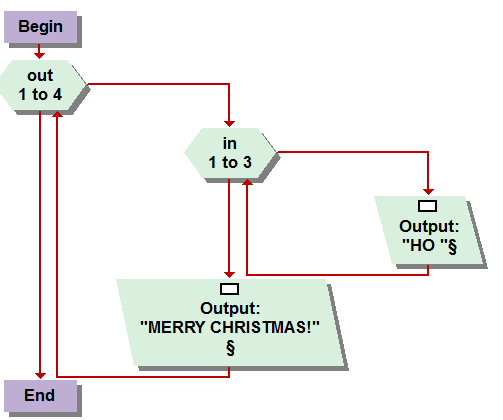


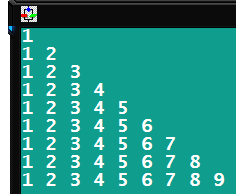
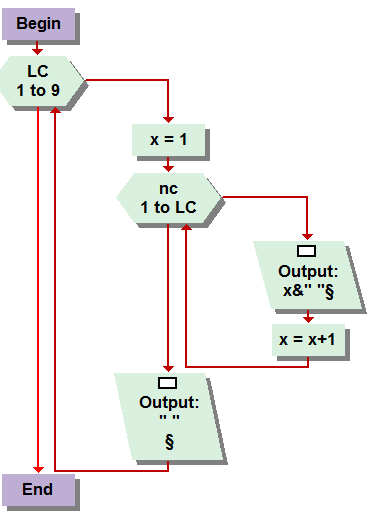
4.1  
An instant classic, BLAST OFF! A very simplistic but ingenious program counting down a rocket launch sequence to blast off. The fundamental use of the necessary FOR LOOP was demonstrated on page 60 figure 4-2. With slight modifications made to the loop itself, inverting it to count down from ten as opposed to counting up to five, and slight tweaking of the text syntax for the assignments sake, I had my rocket blasting off in a matter of minutes.



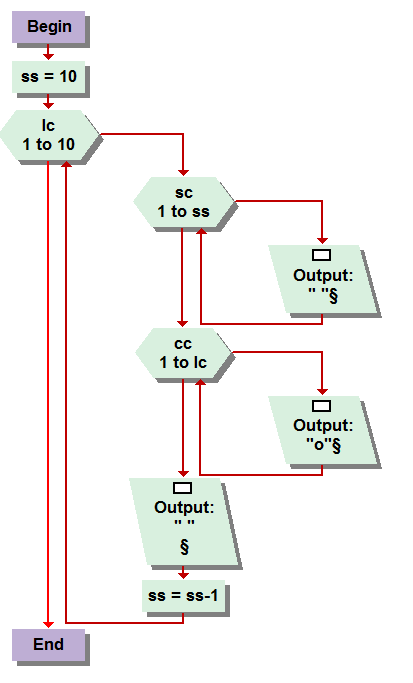
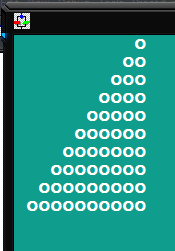
4.2   
I assume root beer is the school friendly approach to this problem, but this is a college course, I think we could have came up with something a bit more mature than rootbeer, although when I do enjoy a root beer, mine comes in a can. The link you sent me in regards to my questions about more complex output statements from the chapter one’s homework came in most handy for this problem. As well, figure 4-5 on page 62 gave me some insight on the algorithm on whole. I basically mimicked that program with again, just the slightest of modifications to the actual loop and output displays to confine to the task at hand. I thought this gave great insight to the practice of creating very eloquent outputs from direct contributions of the console function.



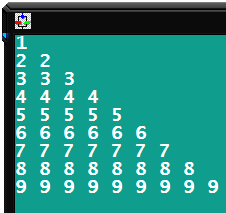
4.3  
The first use of a nested loop, sensational! Figure 4-9 and its output in figure 4-10 really made this one way to easy to simulate. As well 4-11, witch was more word interfaced than the numeric values of the other two examples. The idea was to have your out put display “ho ho ho mary Christmas” 4 times with only programing the word ho once. So you simply assigned it to run a “ho” for loop three times before running the “merry Christmas loop once, with layering in the ever tricky end of console symbol after a space break, or enter key” to display them on separate lines. BY the way, I fixed my JING =D

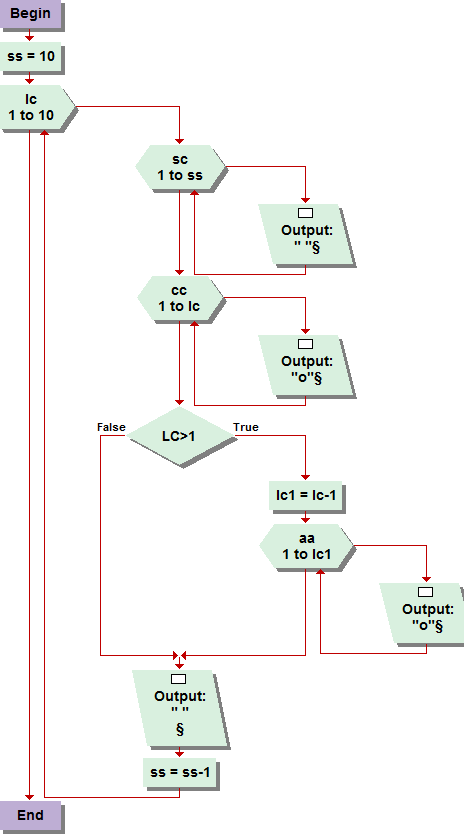
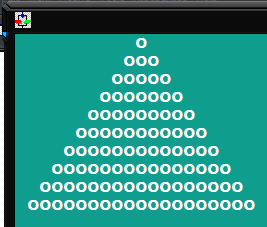


4.4 Now this task gave me a slight hiccup momentarily. I actually realized that you had to reset my “x” variable to one at the beginning, or pre test, maybe? I didn’t quite understand those terms and will re read them soon, for I have a feeling they will be coming in much need later on. Anyway, I instructed the loop to reset it to 1 each time for without that my loop sequence kept adding and adding and wasn’t working. Figure 4-14 came in most handy for this one, as with the creation of the triangle shape, then it just took some simple mathematical calculations, of a very minute nature, to create the output of the correct numbers.

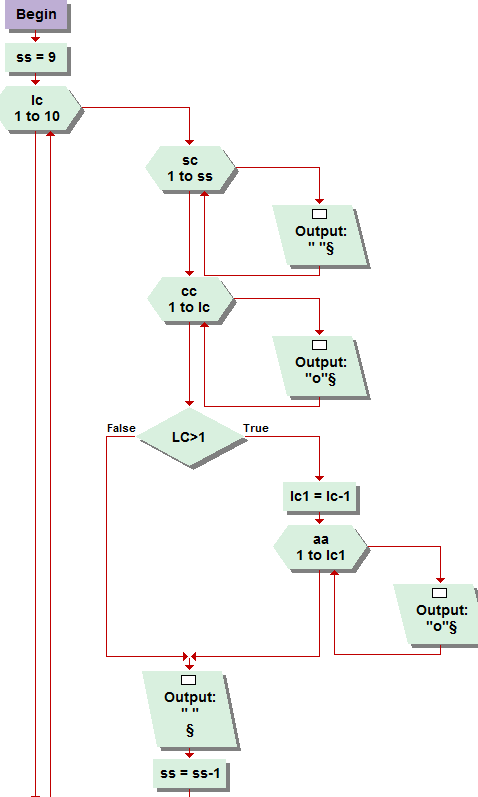
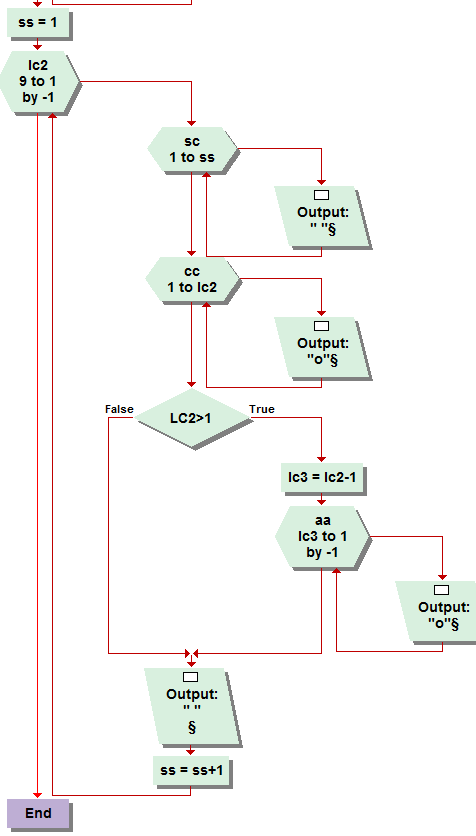


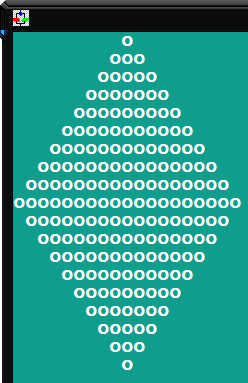
The following I, after further review of the week 4 course document, realize are not required of this week’s assignment. Regardless of whether we need to complete them in the future or not, in case we don’t I feel like I need to submit them for your approval considering I have already done them and, even if I receive no credit, don’t want to feel like I completely wasted my time, which they did infact require some.   
  
4.6 this one was pretty easy to work out, and layed the floor plan for the problems to follow. Figure 4-14 layed the ground work, then it was just a simple process of including a space counter , as mentioned by the question itself, to create the appropriate distance between the circles causing it to appear similarly to the picture in the document , but as if seen in a mirror.





4.7  
This one really got the noodle working. It required many loops as well as a if condition nest all inside a line count loop after u assign a value to ten, ss, which I cannot remember what it stances for, I think its suppose to represent Sircle, lol spacer. Haha but anyway you have the same basic formula for the previousta, but inorder to create the tree shape, you create a loop to form figure 4-14 pattern only if the value of line count is greater then one, this also allows all the circles to exist in an odd number of circles per line. Which, I will admit, I had trouble doing at first. All of these problems took some serious playing with, getting poor results, and re-tweaking, before they came out correct.





4.8  
The ultimate mind blower. This problem took me a long time to figure out, because I made foolish typo’s . I had it right from the start, with converting and inverting each set of loops to form each quadrant of the diamond. Started with the tree formula, inverted and converted for the first quadrant but had a typo for the bottom right quadrant, mainly because I was lazy and used similar abbreviations over and over and got myself confused and had the wrong assignments and loops converting the wrong set of variables. I don’t know exactly how to explain it. Its really quite complicated. And required a good 45 minuets of serious dedication to uncracking. Manily through trial and error, I was able to work out the kinks in conversion and inversion to create the desired result. It was a good deal of fun and I am rather quite proud that I figured it out.