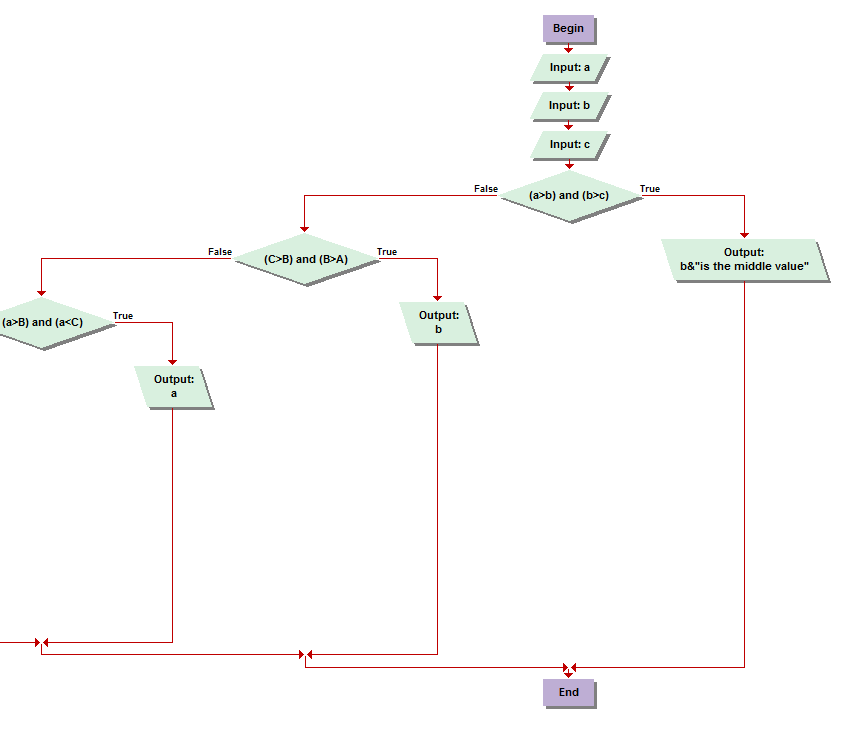
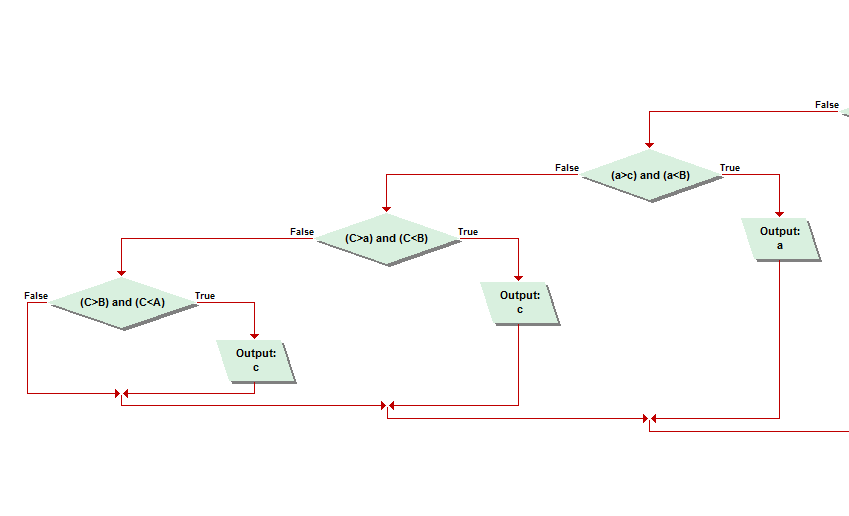
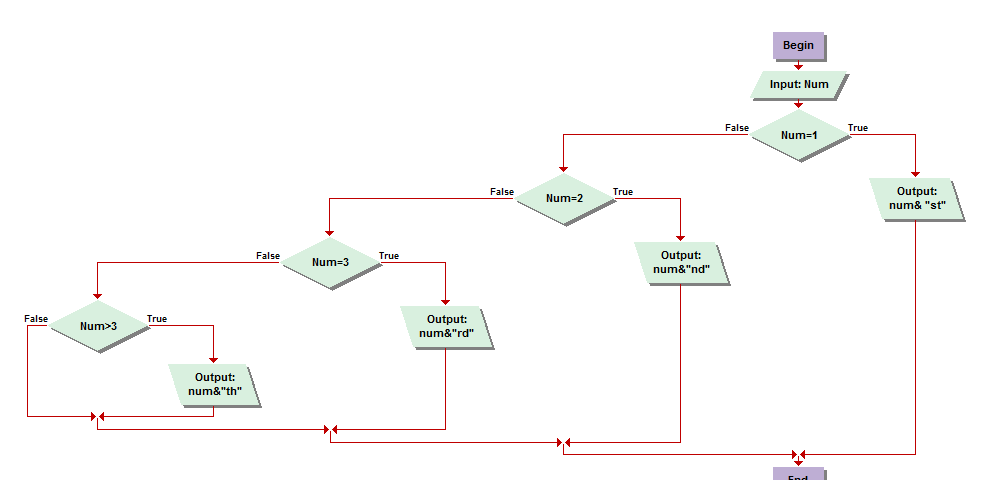


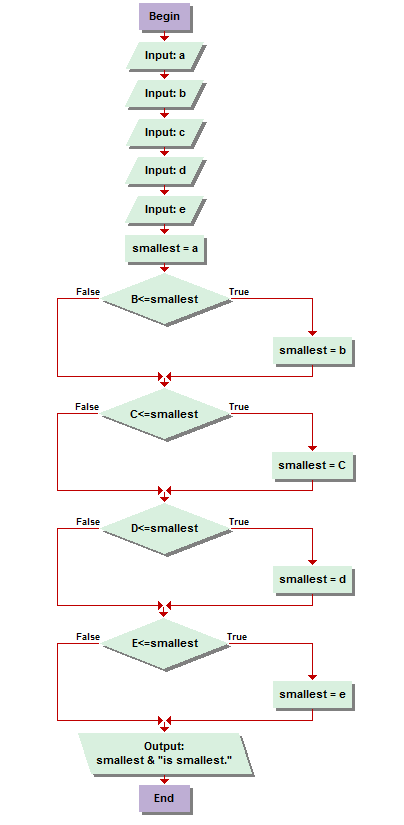
I would just like a bit of preface to my home work documentation before I begin. It is rather late and I started a new job this week. I have already worked over 45 this week alone, its only Thursday, I have a ten hour day tomorrow, and my two year old is sick.   
ANYWAY…  
  
2.1 So for this problem I read on the message board, credit Theodore, before I even re-read the problem in the book to actually start working on it, about multiplying by negative one. I may have thought of that on my own but I didn’t have to since he posted it before I started. A positive minus a lesser positive is always a positive and any negative number multiplied by negative one is its positive value. I’m not sure if I have the energy for charts yet tonight, we’ll see!



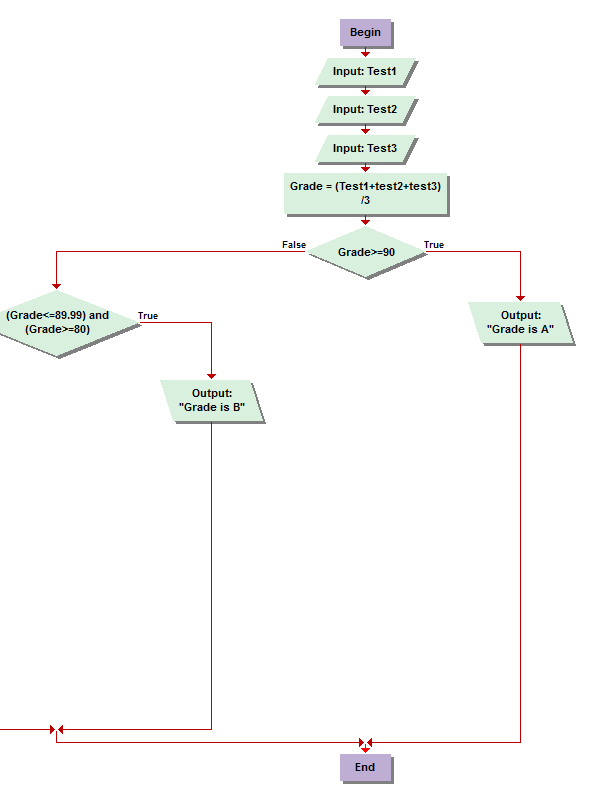
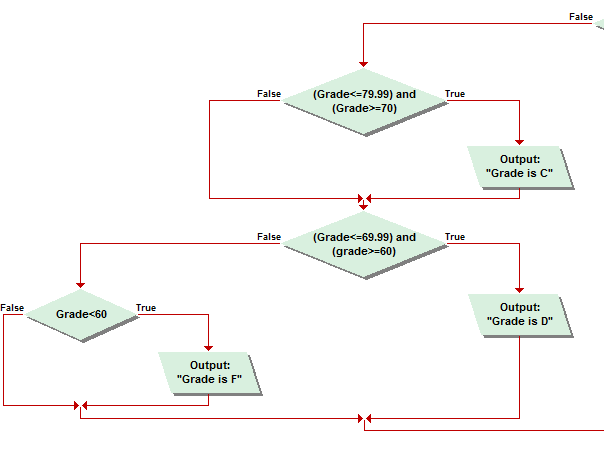
Okay So my Jing is not working properly. Some type of resolution issue in which it’s only enabling me to capture a portion of my screen rather than the entire one. Basically, I had to go old school and use prntscrn and PAINT for my photo editing and it isn’t that great. I feel it is suitable, of course, only if you don’t mind using your imagination a little bit. Just because I couldn’t fit the pictures on the same page without them being so small you could never read the boxes. You will see worse demonstrations of this soon!   
2.3 Trying to find the middle value of three numbers was more difficult than I expected. I scribbled on paper for hours trying to find a pattern between values of numbers and letters representing numbers, i.e.; 1 2 3, 1 3 2, 2 1 3, 2 3 1 … and S M L, S L M, M S L, M L S, over and over till I realized how critically overthinking the solution I was. After countless errors and no out put statements and just no luck at all something just clicked in my mind on how to correctly sequence the conditional statements forcing the application to create output the middle value. At first I do recall trying to only use the same symbol, being either only using < or > and following a pattern but I realized that without ever using a combination you couldn’t deduce the middle value for sequences where you entered the value in any other order than smallest middle large, or vice versa. Product of my own genius!



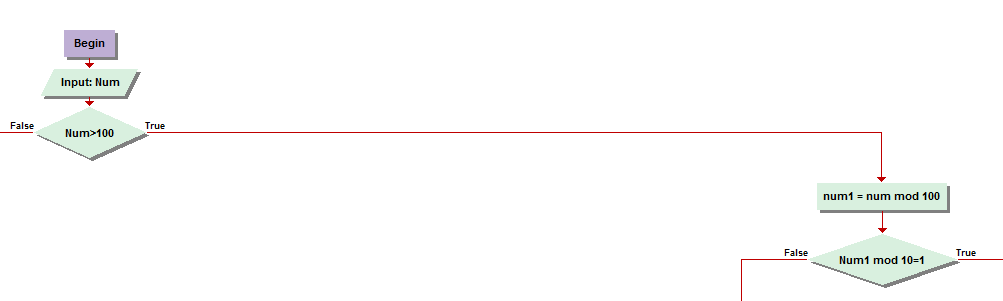
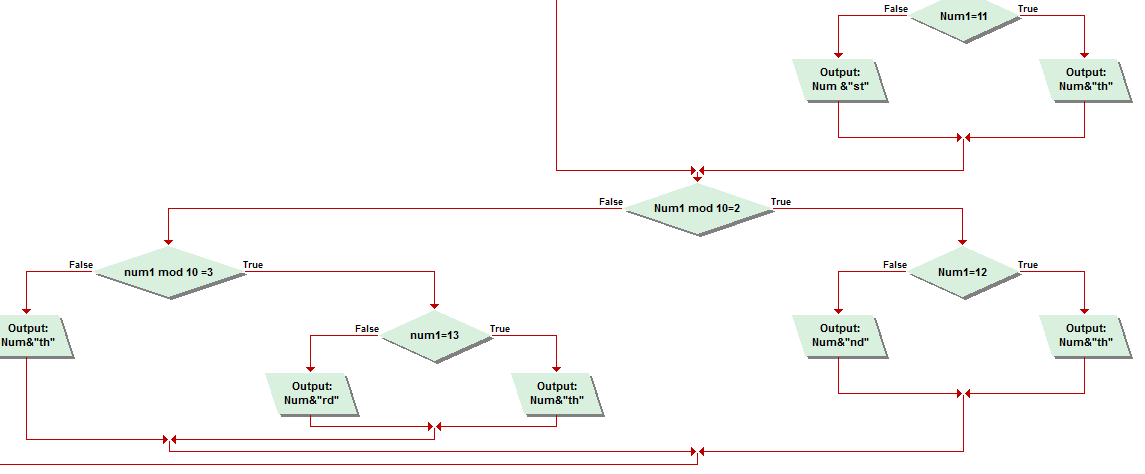
2.2 This one was rather simple, and unfortunately did not give any insight to its predecessor. If the value entered between one and ten was one, two, or three, I used an “if conditional” command to assign the appropriate ending. If the number was greater than 4 I simple assigned an output for “th.”



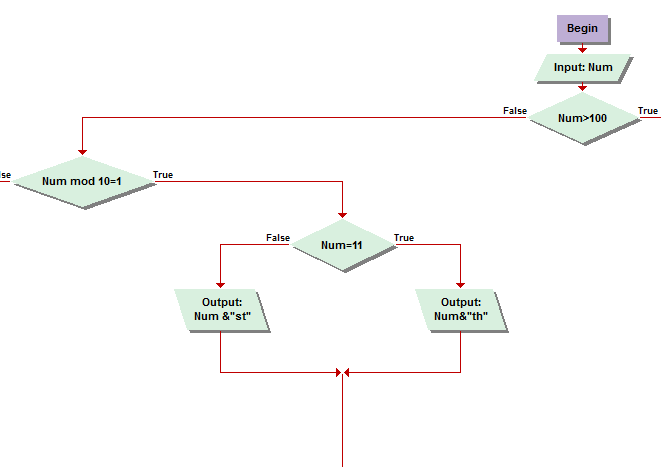
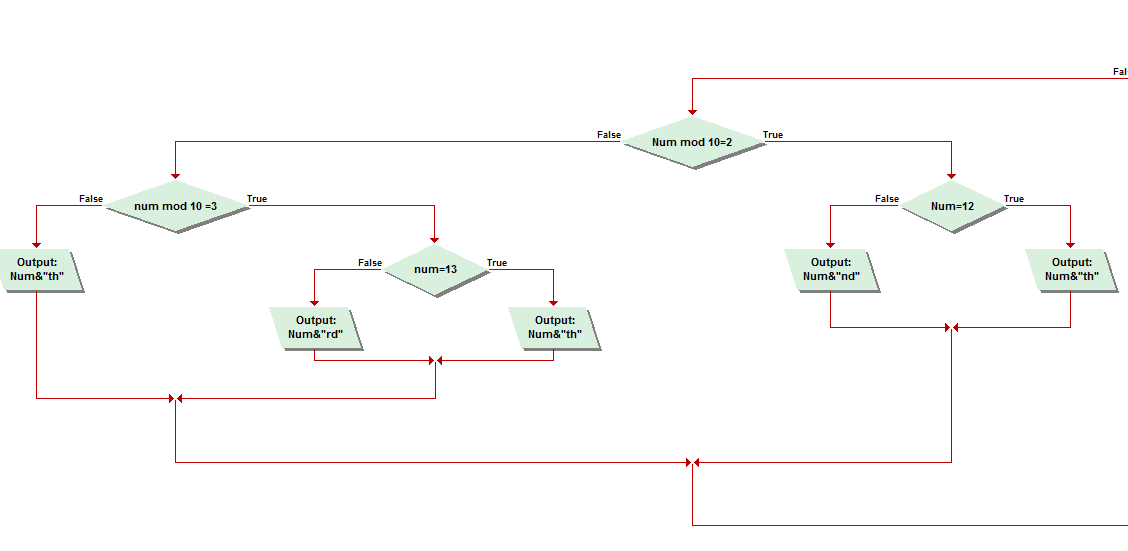
2.4 I found this problem to be a neat variation of the demonstration of the practical display of conditional statements in the book. I found, very quickly through my own deduction and logic, that you could simply add more input values and use a less than or equal to sign for the simplest of modifications to the last demonstration of the smallest value finder in the book. So that is what I did and it worked every time I tried it with all types of different numbers.



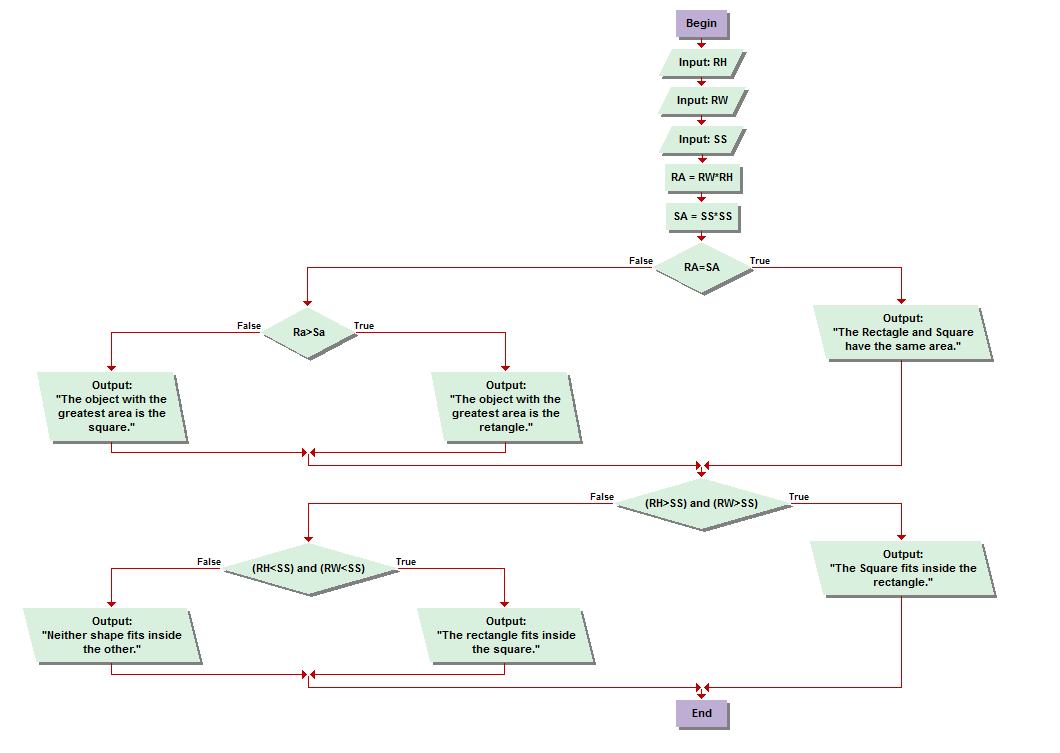
2.5 The grade calculator I found rather simplistic again. Three test score inputs, add them together and divide by three for the average. Soon after, I just used a conditional statement assigning a letter grade to the corresponding numeric value. As I posted on the message board, I did run into some trouble on my first run through with the lower letters finding that I had programmed it under a certain condition, reliant on where I had placed my out put assignments, to show both a C and a F. I fixed it on my next attempt and it worked for the few times after I test ran it without any problems, calculating every letter grade.



Again, with no working jing… had to improvise with the pictures to display the information.   
2.6The two pictures above this are of the right side of my algorithm. If the number inputted is greater than one hundred, the first assignment is to take the number and mod it by 100. The result will be refered to as NUM1. You then follow the same procedures I demonstrate in the left half of the algorithm, which you will find somewhere later in this document. You then mod NUM1 by 1, 2,3. If it mods by those, and the result falls within the teens, it is assigned to add the th ending, otherwise, it receives the corresponding st,nd, or rd. Last you assign the output that any number that doesn’t mod also receives the TH endind.   
  
I must admit that I had a great deal of contemplation, when reading the message boards and finding people who claimed to have solved it, to copy exactly what they display. And honestly, I did. Although I found their charts not to work at all so, after seeing their shortcomings, I was able to piece enough of the good ideas from their improper solutions, combined with a bit of my own ingenuity, into something I tested and even showed my girlfriend, who did not care at all, what I felt was a stroke of genius. I don’t know exactly all who gave me the basic idea on how I approached it, but in the end I scripted the most important and finessed details myself and am rather quite proud of it. But in all honesty it would have taken much less time if I hadn’t read the message boards before reading the and attempting the problem.



2.6 continued. Here is the left half of the algorithm, intended it assign number endings for inputs of numbers less than 100. Basically, its is exactly the same as the right. In fact, I programed this half first and probably should have displayed them in the other order but this is how it came out in word, lol. To be perfectly frank, I made a great deal of use out of Visual Logic’s copy and paste functions when reprograming the right side from the left. This side came first, for finding endings of values less than 100,and then the formulas are reused after “mod”ing the value, if it is over 100, by 100, making it no longer over 100 and practical to run through the algorithm. Some of the rough idea was conceived by various other designs I viewed on the boards, but ultimately after I got a vague idea I did it myself.



2.7 The final problem in the book, at first read, sounded rather tricky, but it was no contest after a second read through. I immediately grabbed some scratch paper and a pen and started scribbling a rough draft. I took into account rectangle high, rectangle width, and square sides. Then, I used two assignment functions for finding their appropriate areas. The conditional function to follow was to determine if their areas were equal. If so, the required sentence was then out putted. Otherwise, they continued through the false arrow to the second conditional function, determining which area was larger, outputting the appropriate deduction. The chart then continues down for the round two of the challenge. Determining if either shape would fit inside either of the other was much easier than it sounded upon reading, as proven by my scratch paper. I was rather impressed by myself not having to actually draw the shapes. First, I used a conditional statement for determining if the rectangle’s height was, as well as the width, larger than the square sides. This indicated that the square would fit inside the rectangle. If both of these were not true, we then determined the opposite of the previous condition. If the rectangle height and width were both smaller than the square’s side, it could, in exchange, fit inside the square. Otherwise, if proven false on all accounts, neither could fit in either.