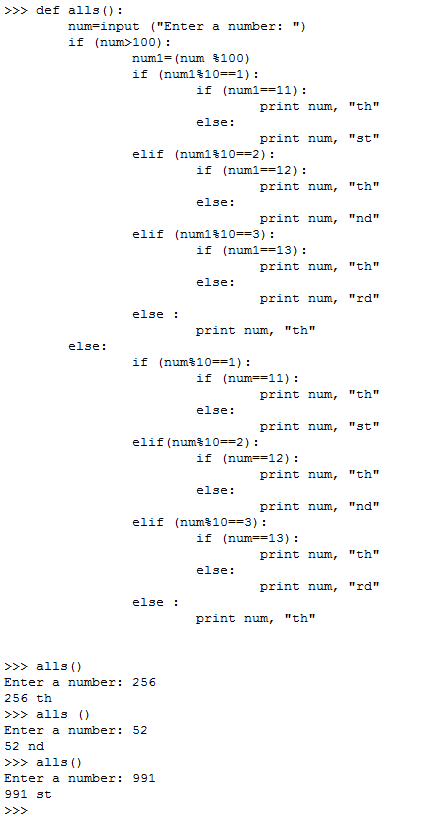
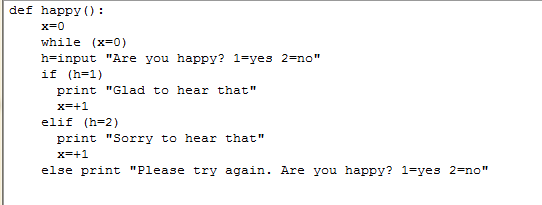
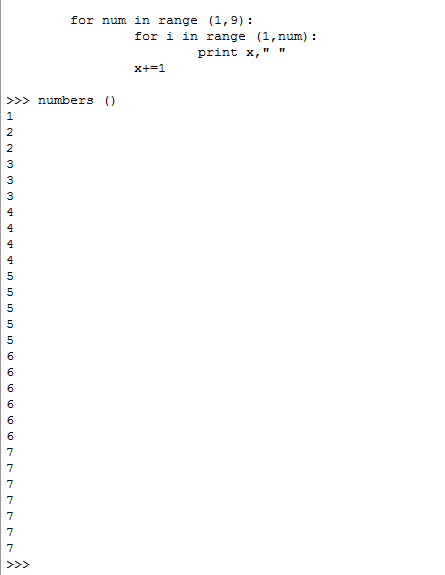
Problem 1.7  
The first old problem we recoded into python was chapter one problem seven, Jake’s problem. Trying to determine how far is car could make it on a full tank of gas. His total tank size is 8 gallons, after driving 60 he has 6 remaining. So we figure out his mpg and multiply it by his total tank size, allowing us to print out the total amount of miles he could travel on a full tank of gas. For this case his mpg is 30 and he can travel a total of 240 miles.   
My car gets about that too!  
  
Problem 2.5   
The grade test average letter score program. This question was a lot less confusing to code in this language, as I found with other ones from this set of questions, because you can eliminate a lot of colorful flow charts with some very short and concise text code.

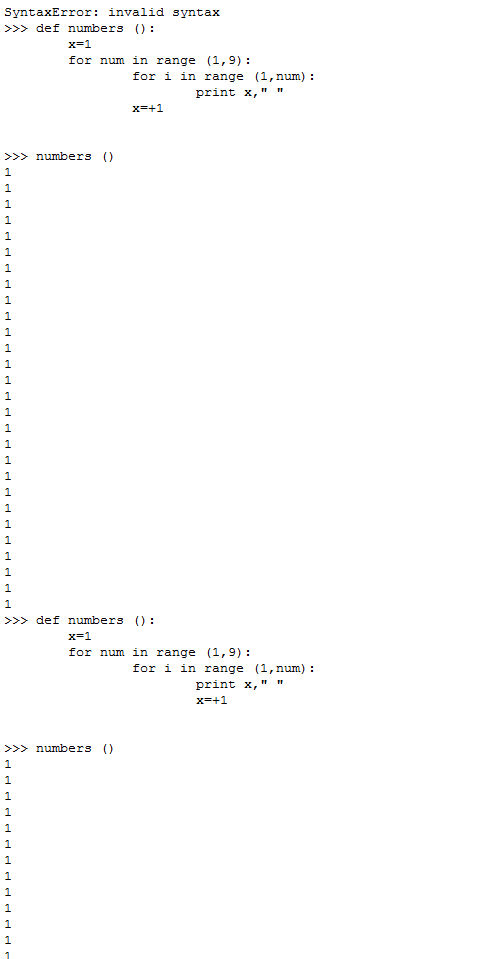
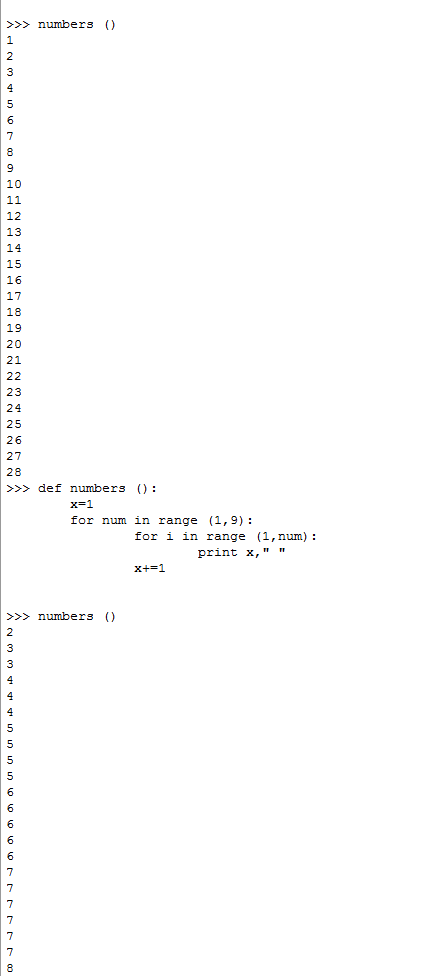
Instead of all the conditional boxes of visual logic, you just have clear exact executions for grading the average of three test scores to a classic letter grade point system used in most schools. In this instance the average is 83 giving it B. With some more intricacy you could even incorporate a + and – system to accompany the grades.

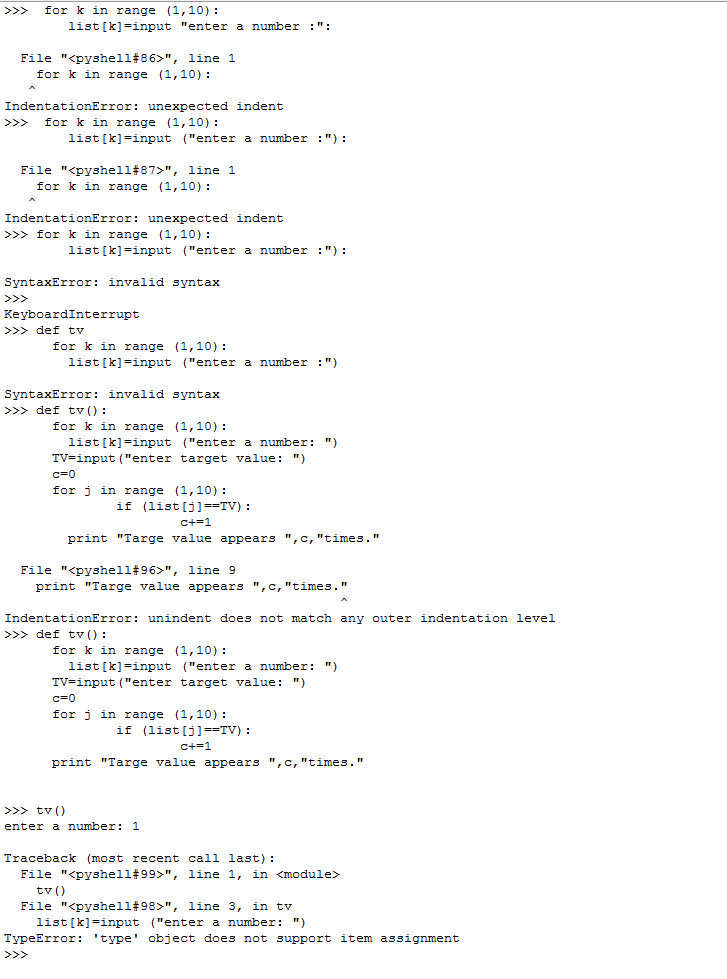
Problem 2.6

This program was fun to begin with, but even better with python. Okay if I thought that the boxes were out of hand with the last problem I had forgot about this one. This VL code had so many if statement boxes and true false arrows all over. All of which was easily summed up with some short powerful python coding. Just some if , elif, and else statements involving using the % for modulus division. Some simple endings for the output and its off without a hitch. I showed a few examples of it working properly on the bottom . Very cool!

  
 Problem 3.5  
Input for Are you happy 1=yes 2=no. Using a while loop on the sentinel condition that x=0, continually prompt the user to ask are you happy and until a 1 or a 2 is entered for yes or no, the sentinel value will only then be triggered. When one is entered, the program will tell the user it is “glad to hear that,” while if no is entered it will reply, ”sorry to hear that.” Otherwise it will say, “Please try again!.”

Problem4.5  
I did not have nay luck with getting this one right, here I got close to it kinda, but had no idea how to grasp the concept of getting the like numbers on the same page to create that triangle shape we produced in VL, the “print” function is a bit more expansive then I comprehend right now and this was my best attempt. It only went up to seven though I coded it for 9, I could see if it only went to 8 but not 7. Some of my prior attempts were not as close, this one atleast had the right number of each number, though it didn’t go high enough and they’re not grouped properly, but I look forward to the homework set next week to see how everyone else accomplished this. My looping skills are not as dynamic in this language as the other one, I hope to see some better demonstrations than the ones I have found online so far.





Problem 5.3  
I had a great deal of trouble getting this one and still not able to complete it right. I hate to admit I just didn’t figure this one yet but the looping properties just didn’t pan out for what I knew on this one. I was trying to reference the value of list elements making them dependent on the current loop but couldn’t figure out why it wasn’t working. The error texts in this language are at times harder to decipher than the code I’m attempting. Sadly I hate to admit defeat on the last two problems of the home work but just a lot going on right now and I have to see some more demonstrations of how to replicate some of these exact procedures and get a bit more practice. It was very enjoyable with the conversions I could do and can not wait to learn to use more of this language.