**Discussion Section Exercises for September 17-18**

Today's exercises involve the `while` statement. The accompanying .py file just contains copies of two of the functions demonstrated in Thursday's class. Your job is to modify them in a number of different ways, described below.

1. The function `one()` prints a list of the squares of the integers 1…10. Modify it so that it prints a list of the same squares, but in descending order---that is, the output should be

   ```
   100
   81
   64
   ...
   etc.
   ```

   (Don't just edit the existing code; write a new function that does this.)

2. We saw in class that if you modify the print statement to

   ```
   print(n*n,end=' ')
   ```

   then the function will print the output as

   ```
   1 4 9 16 25 36 49 64 81 100
   ```

   What happened is that the default string printed at the end of a print statement – normally a 'new line ' character—was overridden and replaced by a space. A student suggested changing the default end to a comma (try this), but this produced the output

   ```
   1,4,9,16,25,36,49,64,81,100,
   ```

   Do you see the problem? It's that last comma, which really doesn't belong there. Revise the function so that it only prints commas between successive numbers, instead of after every number. (HINT: Put an `if` statement inside the `while` statement!.)

3. Now turn your attention to the function `two_b(x)`: This prints a list of all the squares less than or equal to x. For example, if we call `two_b(500)`, the function prints a list of all the squares up through $22^2$=484. Now revise the function so that it does not print anything, but instead *returns* the value of largest integer whose square does not exceed x. So in case the argument is 500, for example, the program will return 22. (HINT. Initialize a variable

`count` to 0, and use it to count the number of times the statements in the body of the loop are executed.)

4. Finally, just so you don't forget how functions interact with the programs and functions that call them, write a main program section that prompts the user to enter a number, and then prints the value returned by the function you wrote to solve Problem 3 above when this number is passed as a parameter. For example, if you type 500 at the prompt, the program should print 22. In fact, this program just prints what you would get if you called

   ```
   print(int(math.sqrt(x)))
   ```

   but I'm asking you to do this in a rather roundabout way.

   The solution is very simple, just two lines of code will suffice. (You could even do it with one line!)