Math 451 ASSIGNMENT 3 **Due at 4 PM Friday, April 4th**

The main outcome for this assignment is to understand how to use a "stopping condition" with a while loop in order to perform an iteration until a desired result is achieved. First we will make a relatively simple modification of your Fibonacci file from Assignment 2 to understand how a while loop works, then we'll do something more interesting with a while loop! One thing that makes while loops a tiny bit more difficult than a for loop is that you need to create a "counter" and update it each time you pass through the loop, whereas this is done automatically for you by a for loop. Another difference with a while loop is that, rather than just passing through the loop a set number of times, after each pass through a while loop a test is done to see whether or not to go through the loop again. In this exercise you'll modify your previous Fibonacci script file to compute terms of the Fibonacci sequence until the first term exceeding a given value (we'll make it 200) is reached.

NOTE: If at any point you try running a script and the command window gives no response and the command prompt >> disappears, your program is "stuck in a loop." You can stop it with *Ctrl-C*. If you ever forget this you can do a web search for "stopping *MATLAB*" and you will find this easily.

1. Take your file from Assignment 2, rename it `assn3A` and get rid of all the comment lines. After the two lines initializing the values of `fib(1)` and `fib(2)`, initialize your counter variable k by giving it the value two. Then change the first line of your for loop to `while fib(k)<200`. What this will do is test to see whether the most recently computed term is less than 200. If it is, the steps in the loop will be carried out. If it is not, the script will "jump" to the first line after then `end` line of the loop.

2. Try running your new script now. Something will go wrong - read the red warnings in the command window, and see if yo can figure out what they are telling you. You will need to insert a line right after the while command in o0rder to increase k by one before computing the next term of the Fibonacci sequence. If you are a programmer you won't flinch at the mathematically absurd statement for doing this: `k=k+1`!

3. Hopefully it worked - did you get terms of the sequence, but with only one over 200? If not, try to figure out what is wrong and fix it.

4. We want to get in the habit of adding a "safety' feature to all of our while loops to prevent them from getting stuck if we make a mistake in our code. Change the first line of the loop to `while fib(k)<200 and k<100`. The choice of 100 is made here in the hopes that the first condition, the one we are really interested in, will be reached before k reaches 100.

5. Change the name of the file to *yourfirstname_assn3a* and e-mail it to me.

Save another copy of the file as *assn3b*. You'll now modify that file to do something more interesting!

6. Given the Fibonacci sequence $a_1, a_2, a_3, a_4, a_5, ... = 1, 1, 2, 3, 5, 8, ...$ we wish to construct another sequence $r_1, r_2, r_3, r_4, ...$ of the ratio of each Fibonacci number to the previous Fibonacci number. So, for example, r_4 would be 3 divided by 2 or 1.5. Now an obvious(?) problem with this is "How do we define r_1?" Well, for reasons that I'll explain later, we'll let $r_1 = 0$. Calculate the first five terms of the sequence of ratios "by hand" (which really means without *MATLAB*, so you can use your calculator).

7. Take your renamed file and add some lines before the loop to initialize the sequence of ratios the same way that you initialized the Fibonacci sequence. (Call that sequence `ratio`). Turn the plot line into a comment and add a line after the loop to to have your file output only the sequence `ratio`, which at this point will consist of only the initialized terms. Run the file to make sure it does that.

8. Add a line within your loop to compute additional terms of the ratio sequence at the same time that you are computing terms of the Fibonacci sequence. Run your file to see if it seems to have worked. What do you notice about the terms of `ratio`? One thing should be obvious, but another is more subtle: Are then terms getting larger and larger, smaller and smaller, or neither?

An essential idea in both pure and applied mathematics is that of finding the distance between objects. The most common way to do this is to have a method for determining the "size" (which is a non-negative number) of a single object, then to say that the distance between two objects A and B is the size of the difference $A - B$. We often denote size of an object A by $\|A\|$, so the distance between A and B is

$$\text{dist}(A, B) = \|A - B\| = \|B - A\|.$$

The fact that the order in which subtract A and B doesn't matter is a result of how we define the size of an object. It is possible to go through all this in a much more formal manner, but hopefully a concrete example will make it clearer. With numbers, the absolute value of a number gives us its distance from zero, which is a reasonable way to define the size of a number. We can then find the distance (on the number line) between two numbers by simply subtracting the numbers and taking the absolute value of the difference. For example, thinking about a picture will easily convince us that the distance on the number line between -3 and 5 is eight units. We can get this by

$$\text{dist}(-3, 5) = |-3-5| = |5-(-3)| = 8$$

9. Now it appears that any two consecutive terms of the ratio sequence get closer together as we go farther and farther out in the sequence. What we want to do now is change to stopping condition of the while loop to be based on the ratio sequence rather than the Fibonacci sequence. Use the above information to make the new condition such that the loop continues to run as long as the difference between two terms of the ratio sequence is greater than 0.0001.

10. Try running the file to see if it works. You may wish to see more decimal places. To do that, type `format long` in the command window, then run your file again. Check the last two terms of the sequence to see if they really are less than 0.0001 apart, and the previous two to see if they are more than 0.0001 apart.

11. When you think your file does what it should, go ahead and remove the comment from the plot line, add your name to the front of the file name (so it will now be *yourfirstname_assn3a*) and e-mail me that file.