**ENR 261 Spring 2023 Chapter 5 Homework**

**General Instructions:**

Save your all your Matlab files for this chapter in the folder named **Ch05** located inside your local repository on your USB Memory Stick. When finished be sure to add, commit, and push your changes to your remote repository on GitHub.

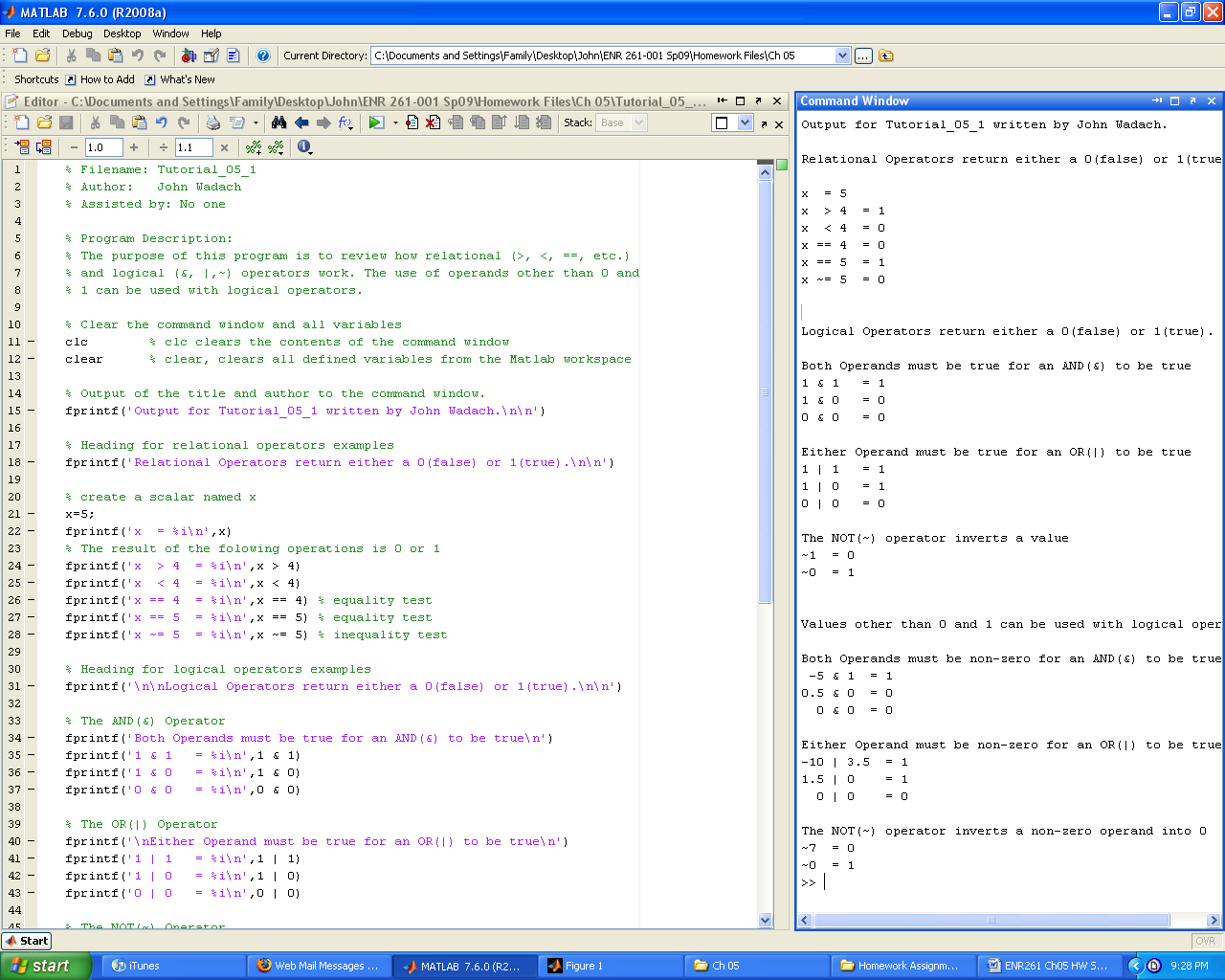
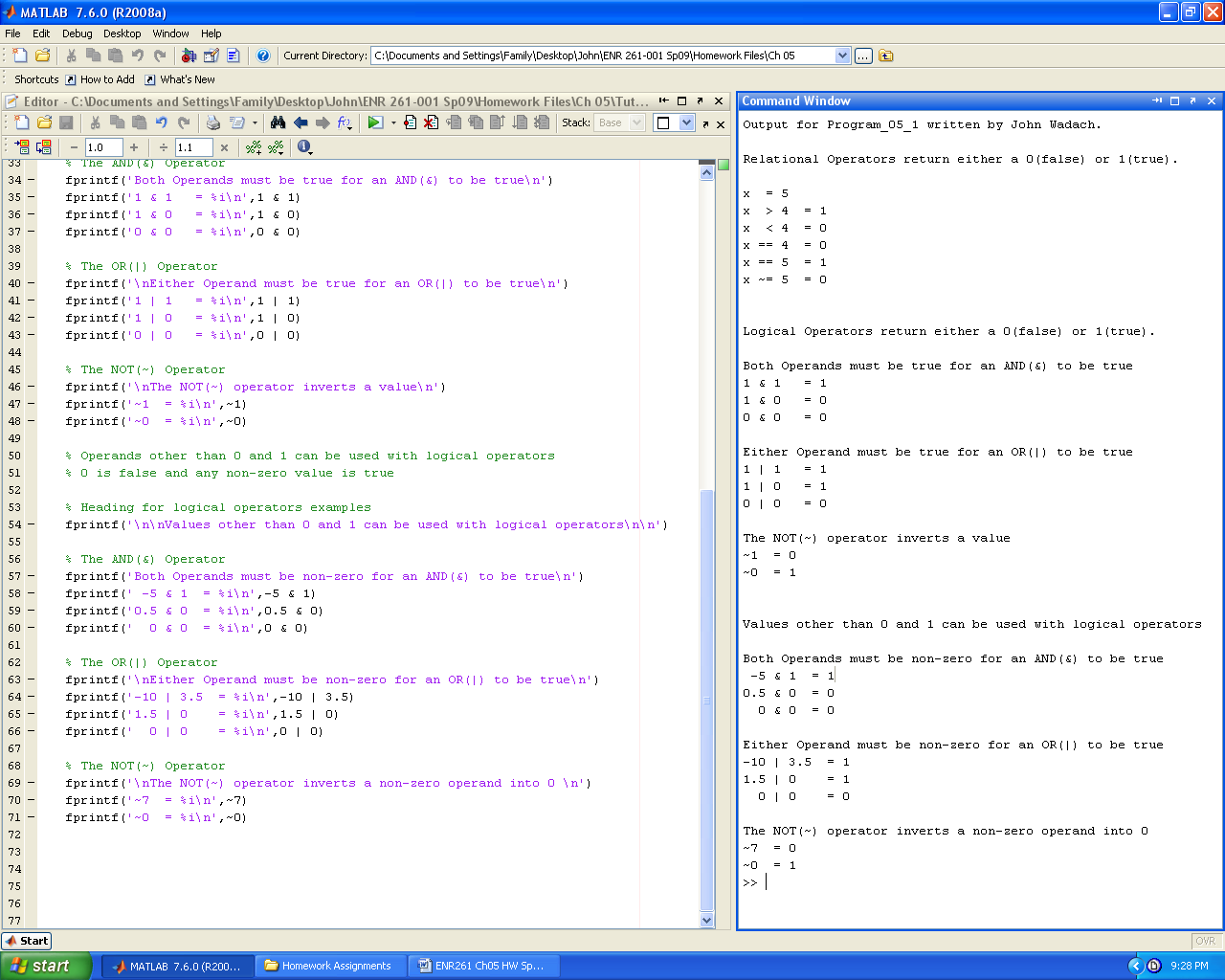
**Assigned Exercises**

1. Recreate all of the following script files and be sure to save them in your local repository on your USB memory stick, commit the changes and push them to GitHub.

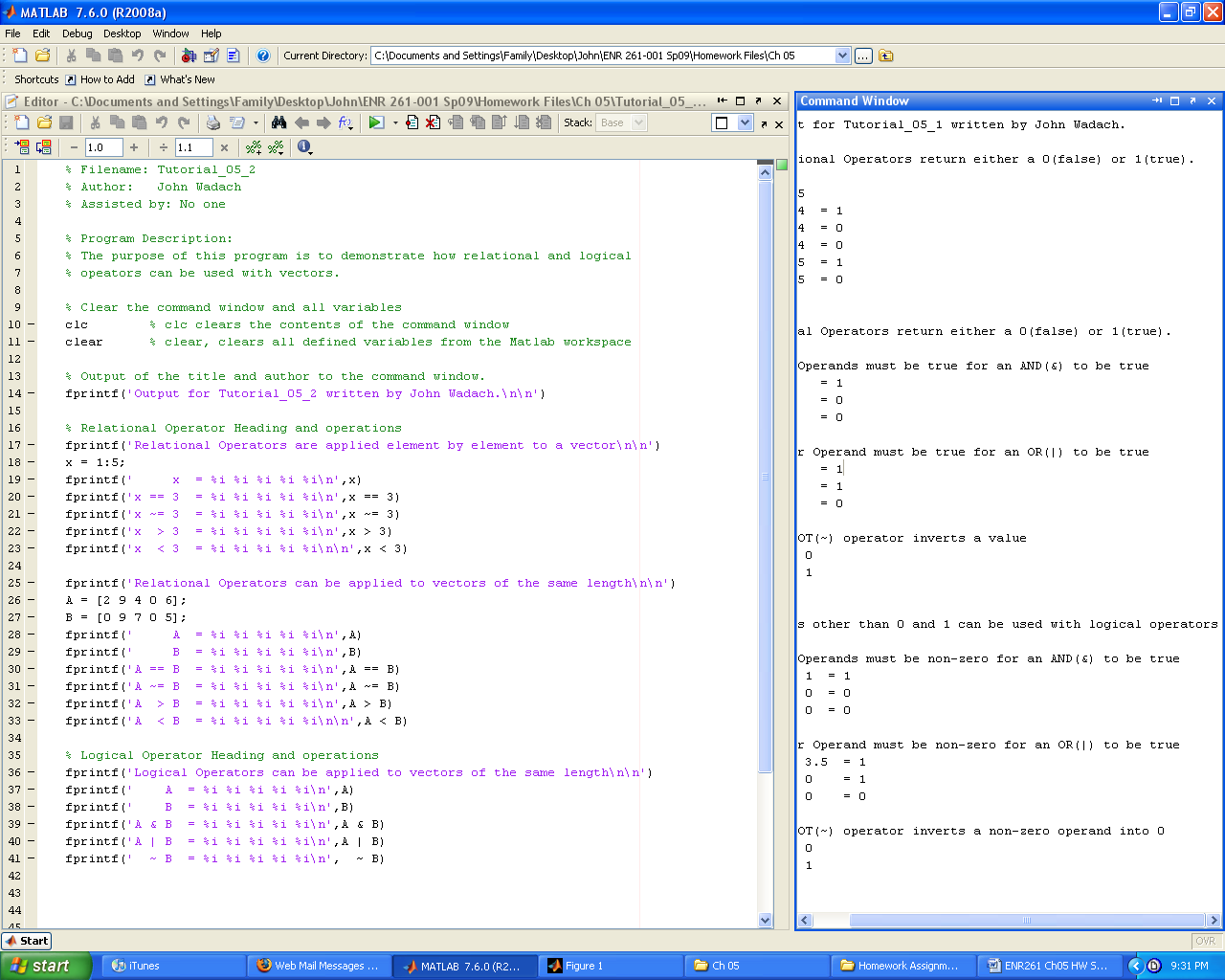
2. Use the required file names for each script file.

Required File Name: **Tutorial\_05\_1.m**

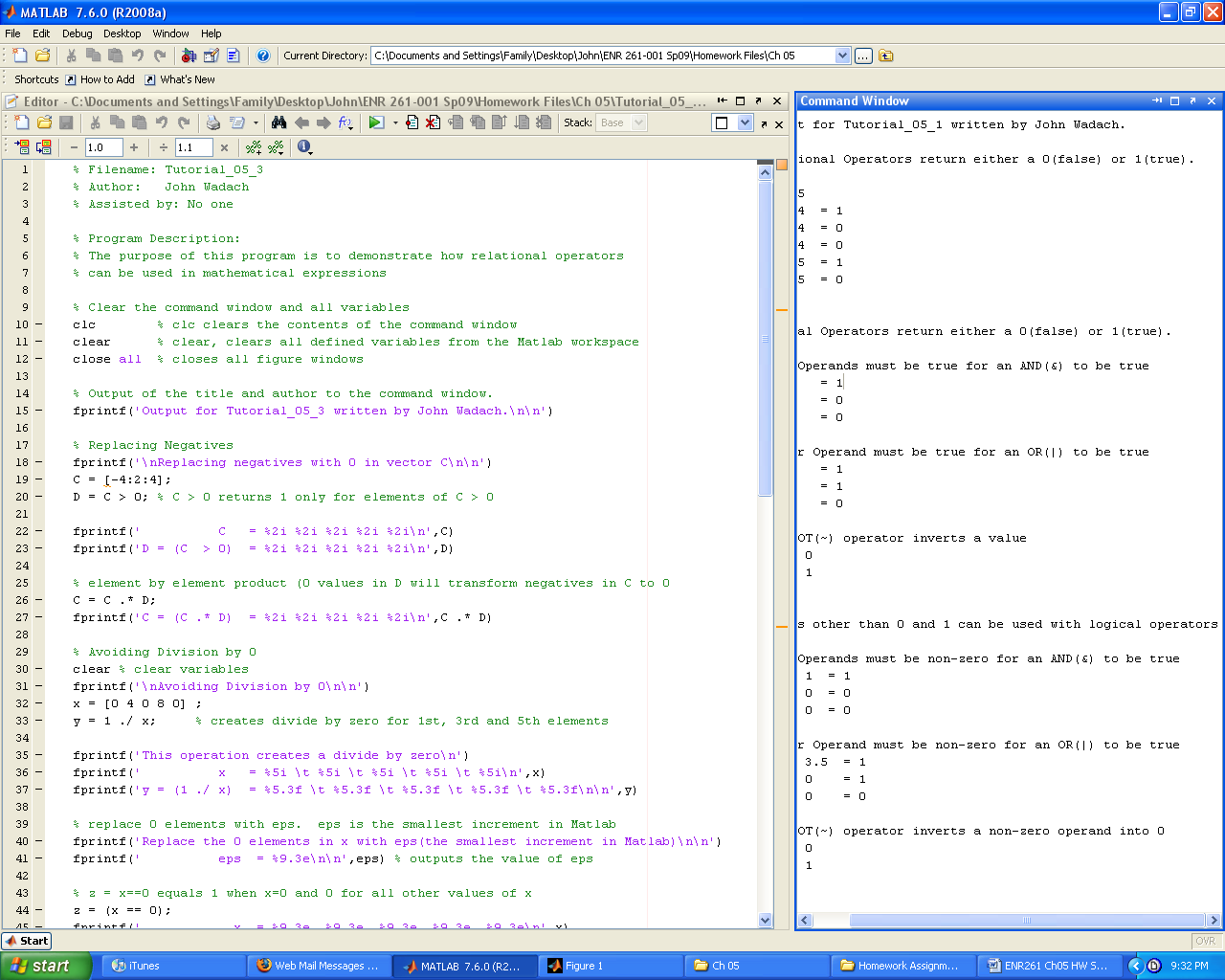
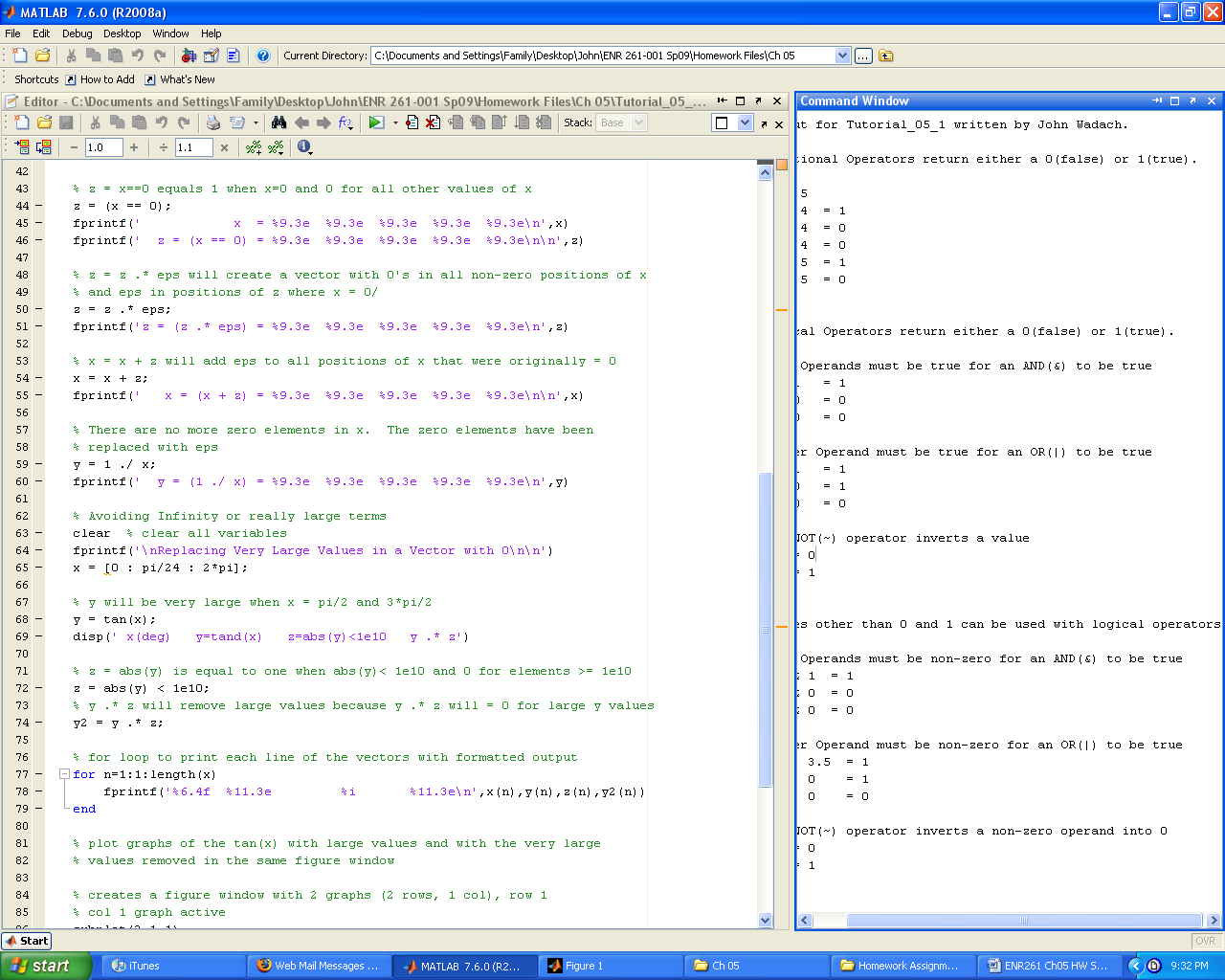
**Continued on the Next Page**

****

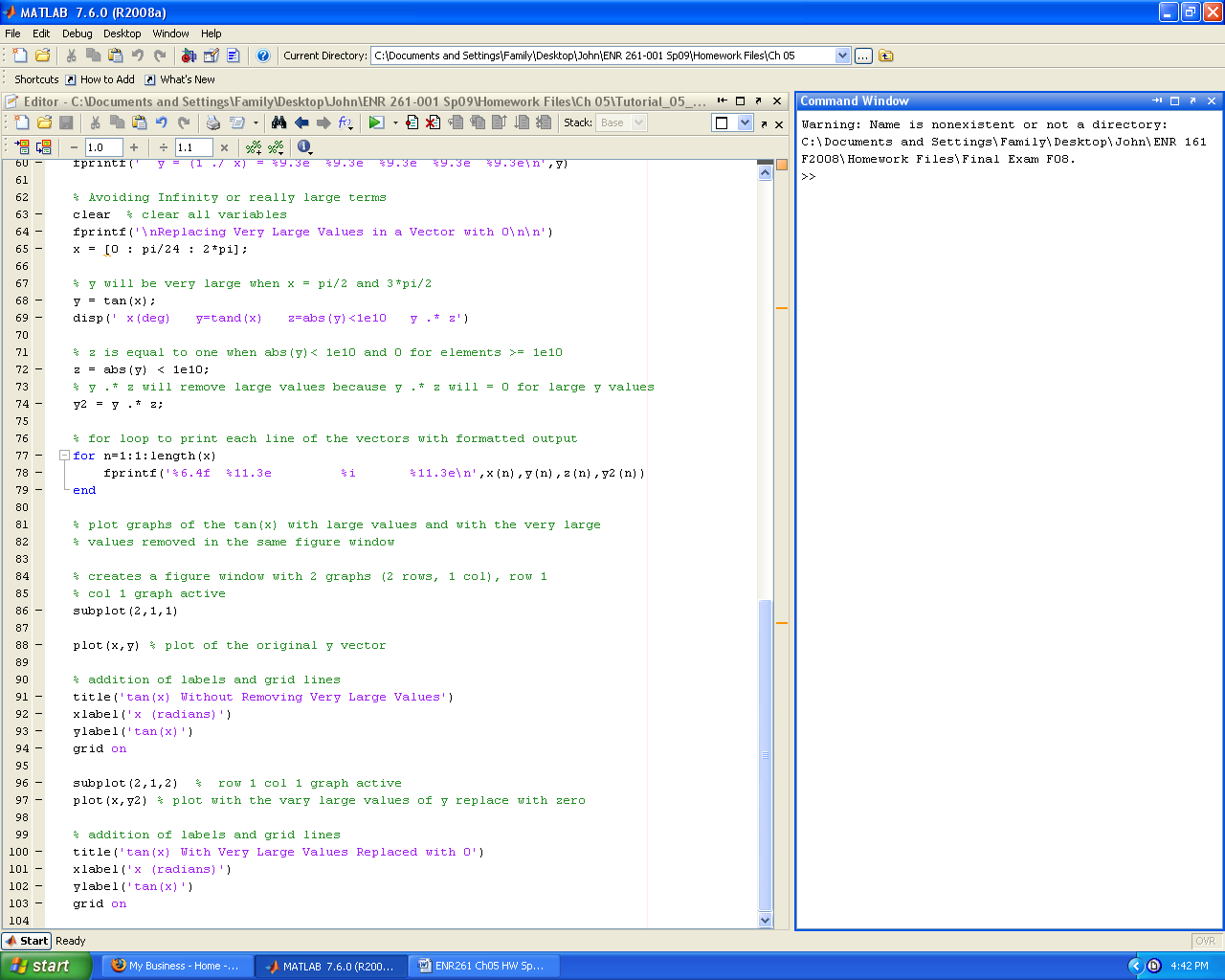
Required File Name: **Tutorial\_05\_2.m**



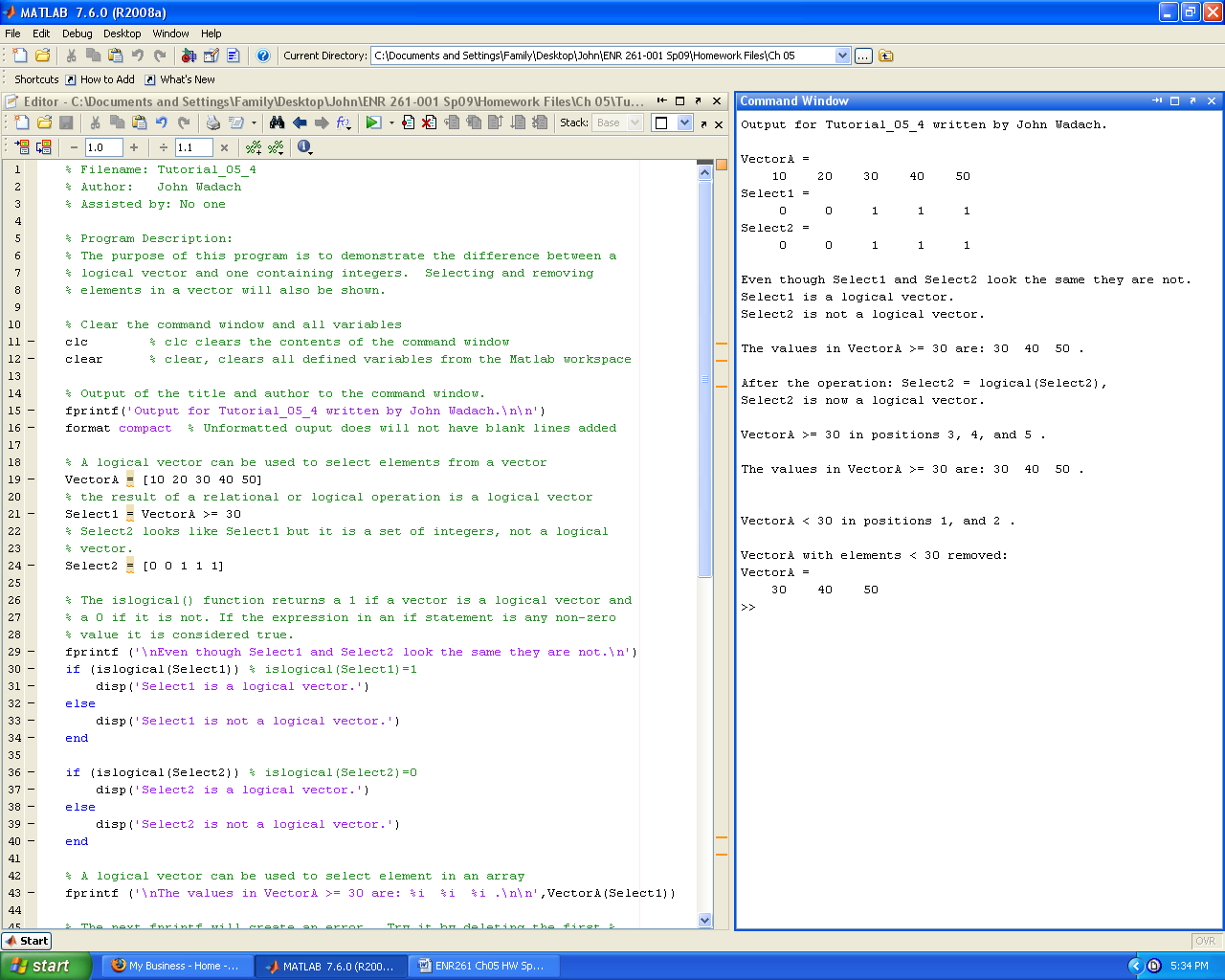
Required File Name: **Tutorial\_05\_3.m**

** **

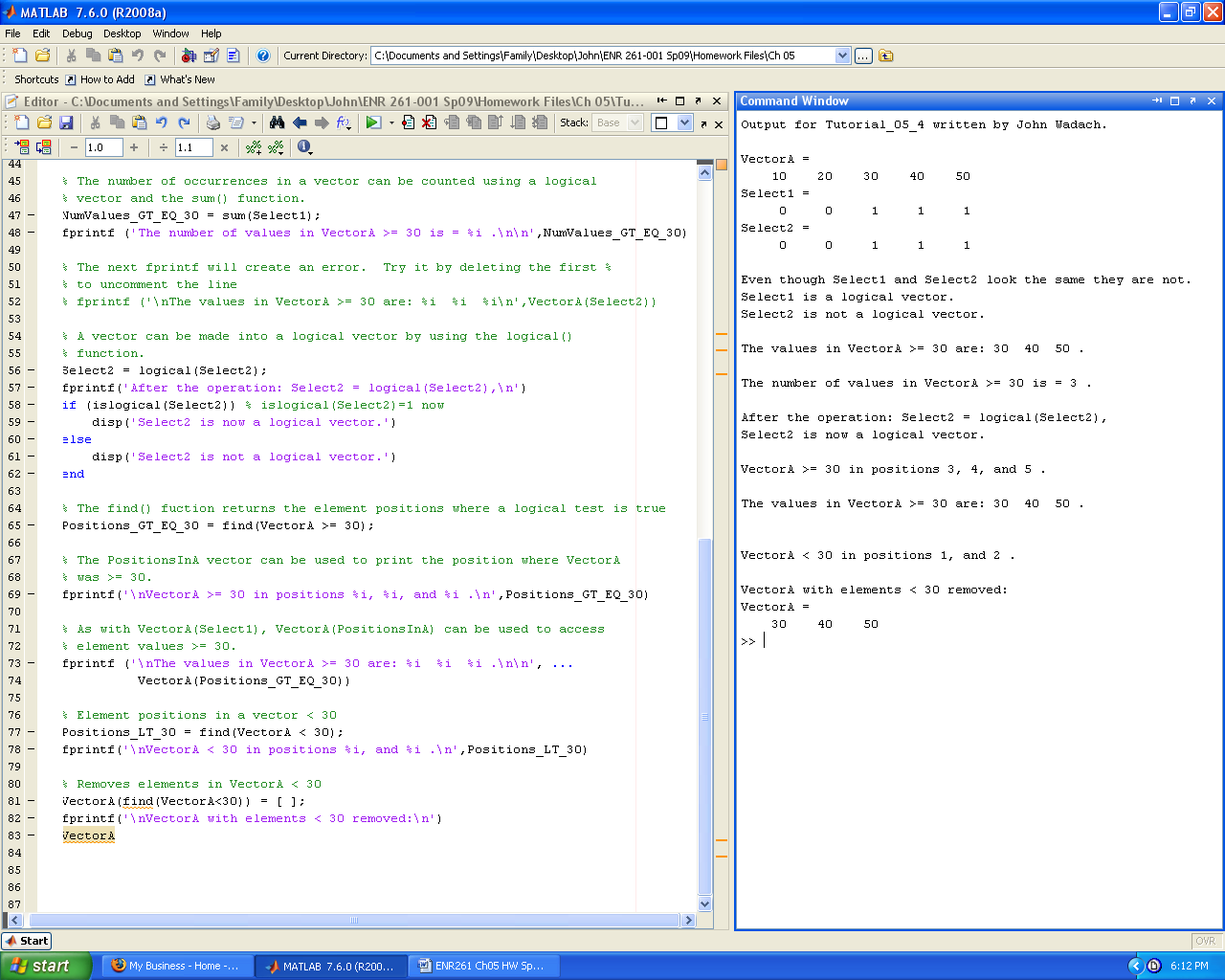
**Continued on the Next Page**

****

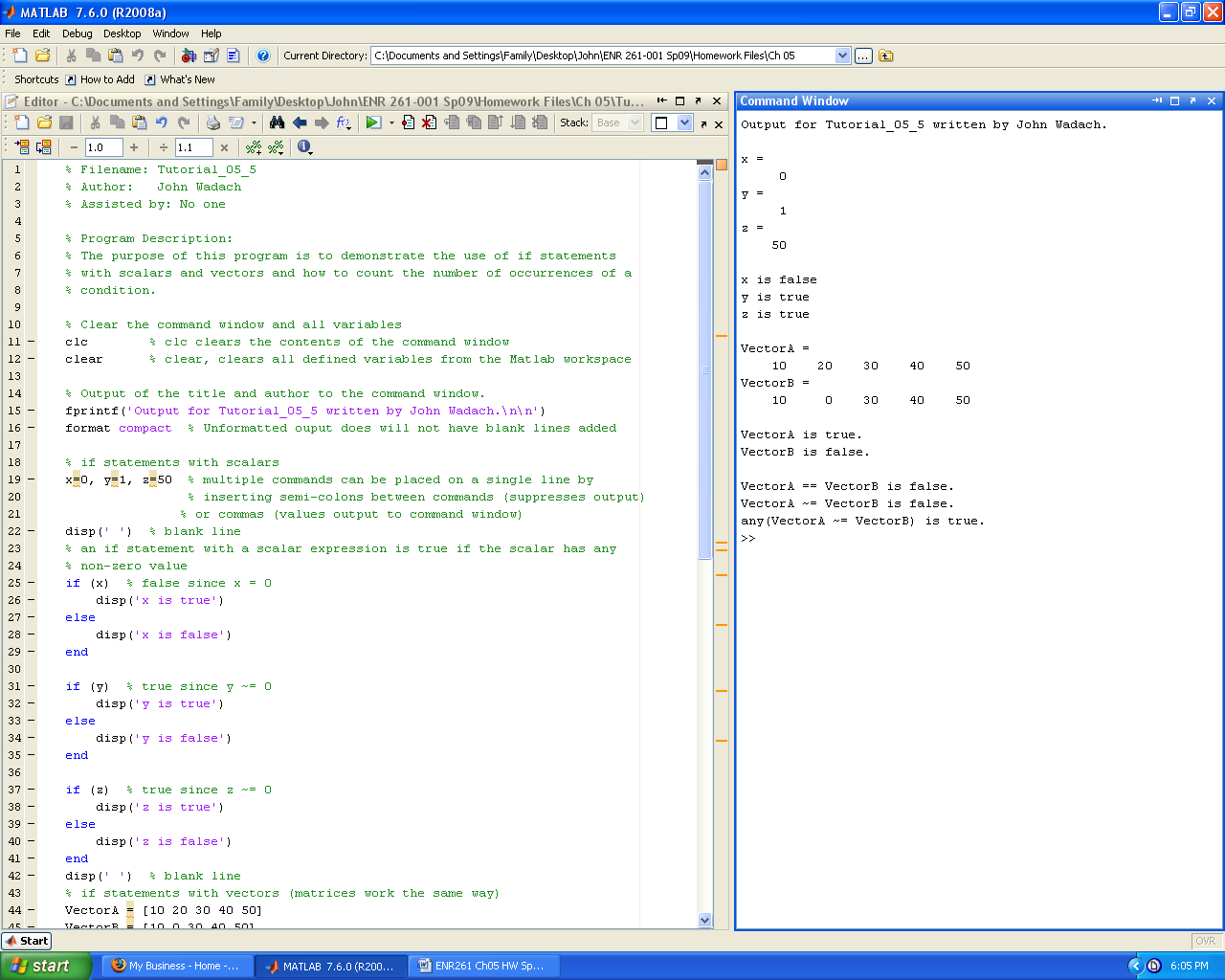
Required File Name: **Tutorial\_05\_4.m**

****

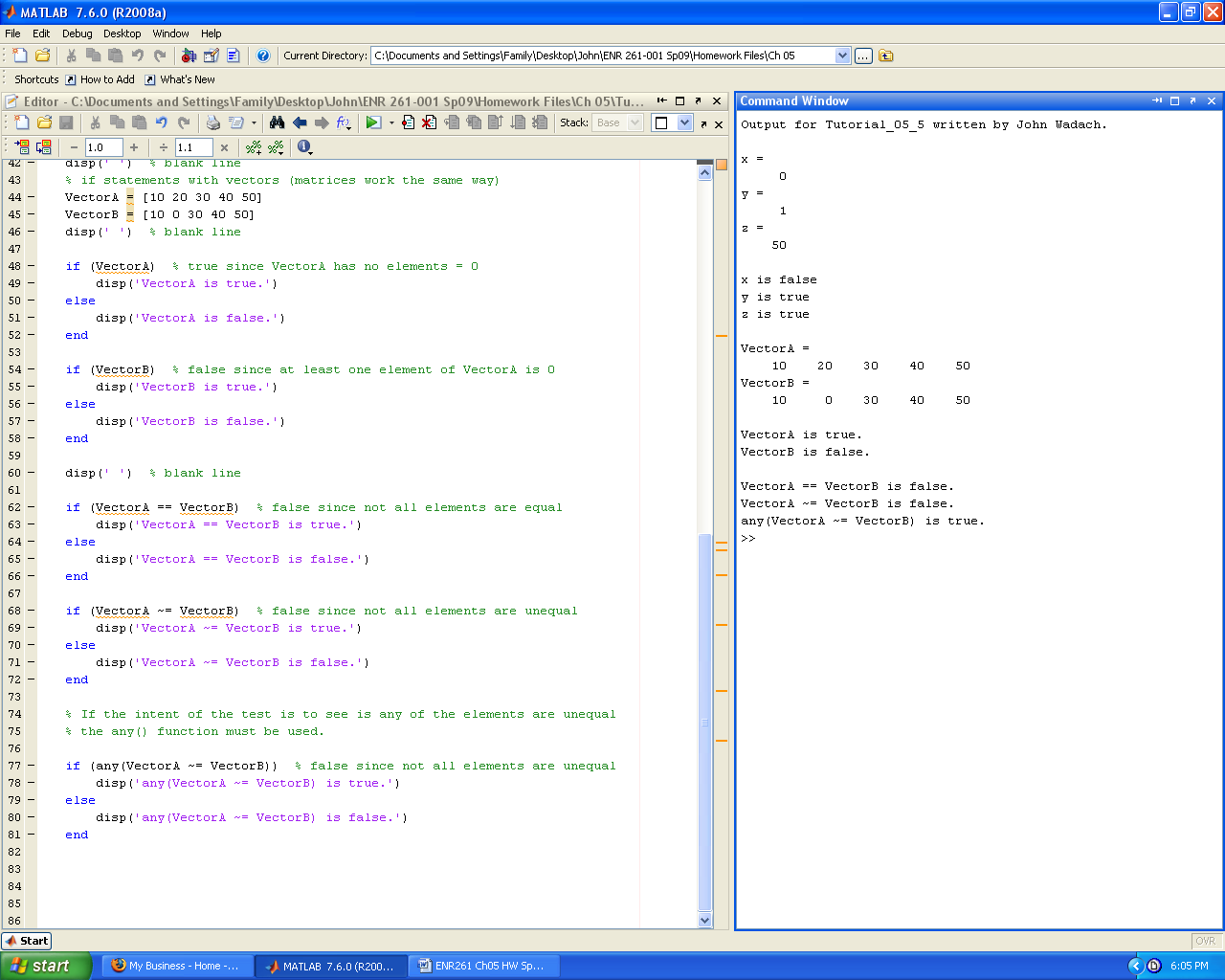
**Continued on the Next Page**

****

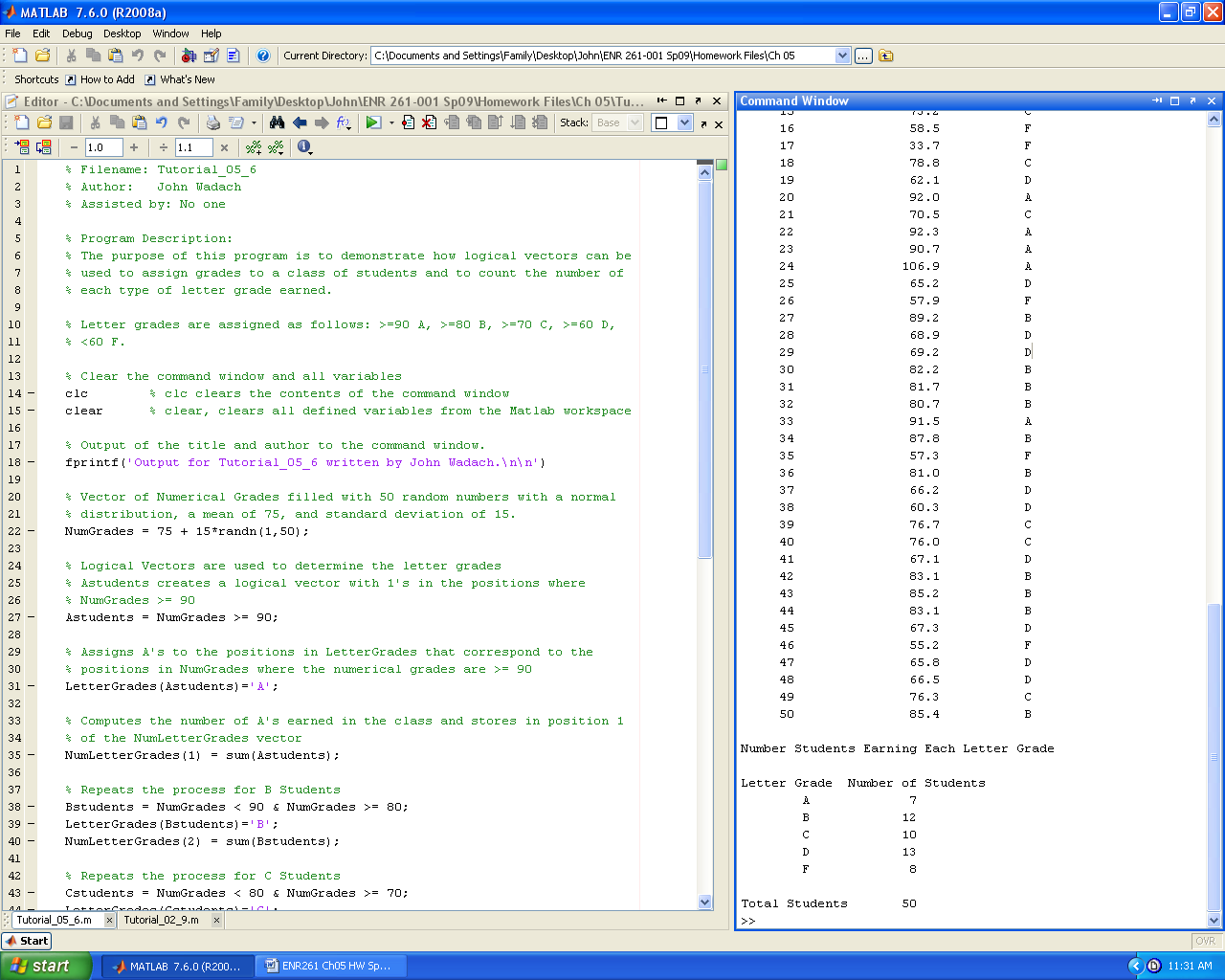
Required File Name: **Tutorial\_05\_5.m**

****

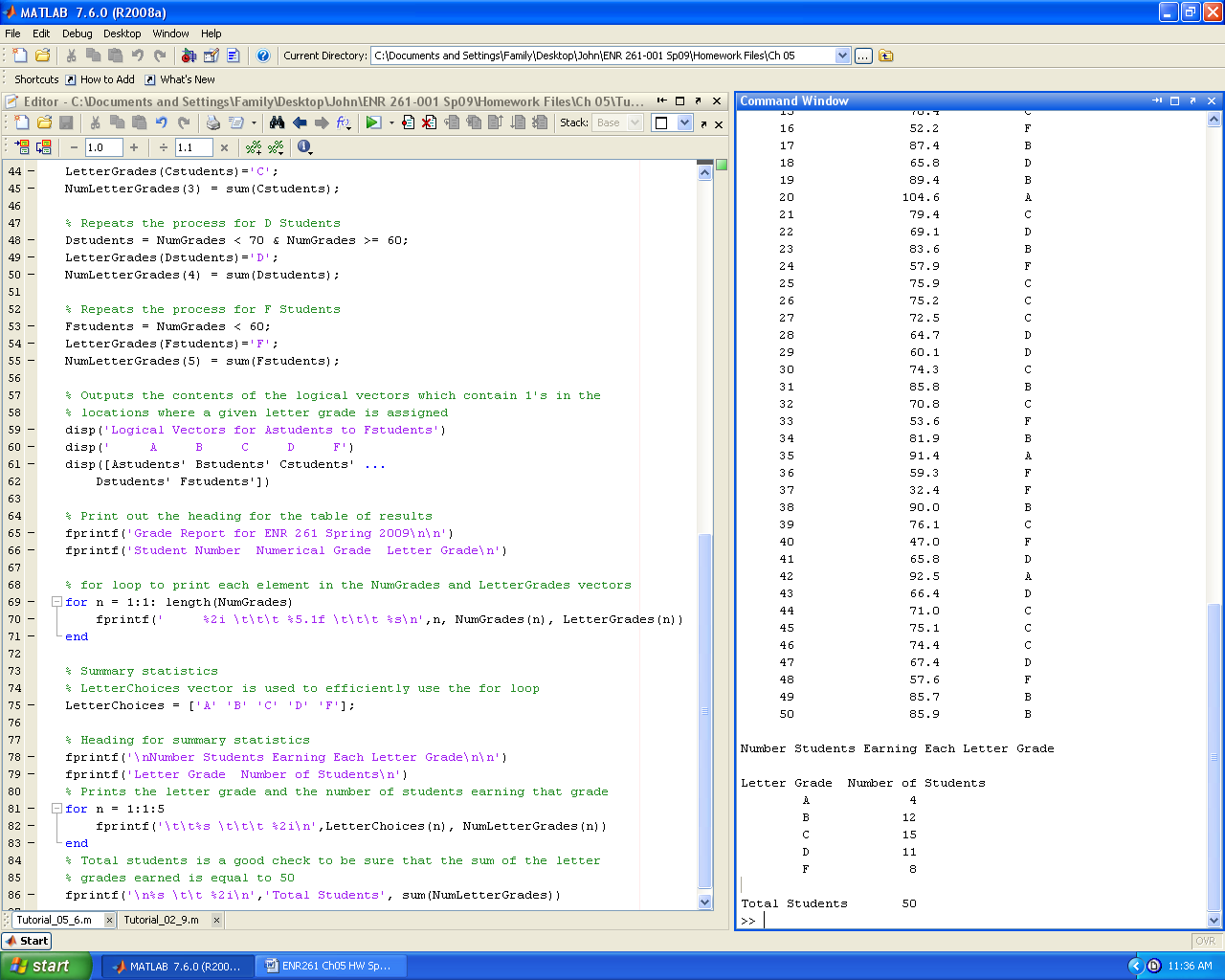
**Continued on the Next Page**



Required File Name: **Tutorial\_05\_6.m**

****

**Continued on the Next Page**

****

Required File Name: **Program\_05\_1.m**

1. For the following, you will use the Excel file: **ClassList.xlsx** found in your local repository.

2. Perform the operations outlined below using Logical Vectors when appropriate.

Program Description:

This program will read in the original data from the Feb\_6\_2009 sheet

contained in the Excel file named ClassList.xlsx . This file contains the

student number and number of absences recorded for each student as of Feb

6, 2009.

Perform the following operations with this data:

1. Output the Feb 6 attendance data to the command window with a neat format.

2. Output a list of students with 1 or 2 absences in the command window

who will be sent warning letters.

3. Output a list of students with >=3 absences in the command window

who will be dropped from the course .

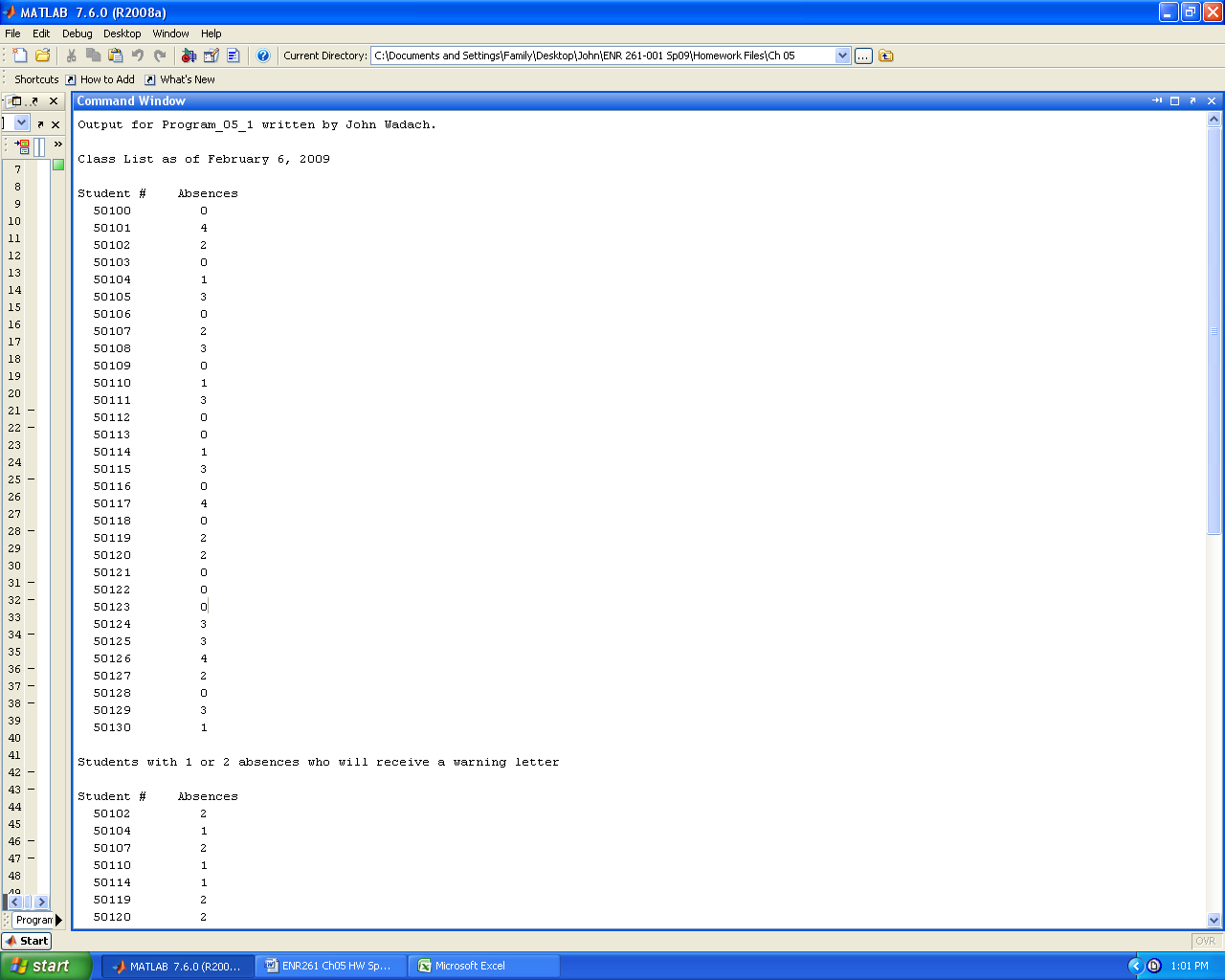
4. Remove the students with >=3 absences from the class list and output

this updated class list to the command window and the Feb\_13\_2009 sheet in

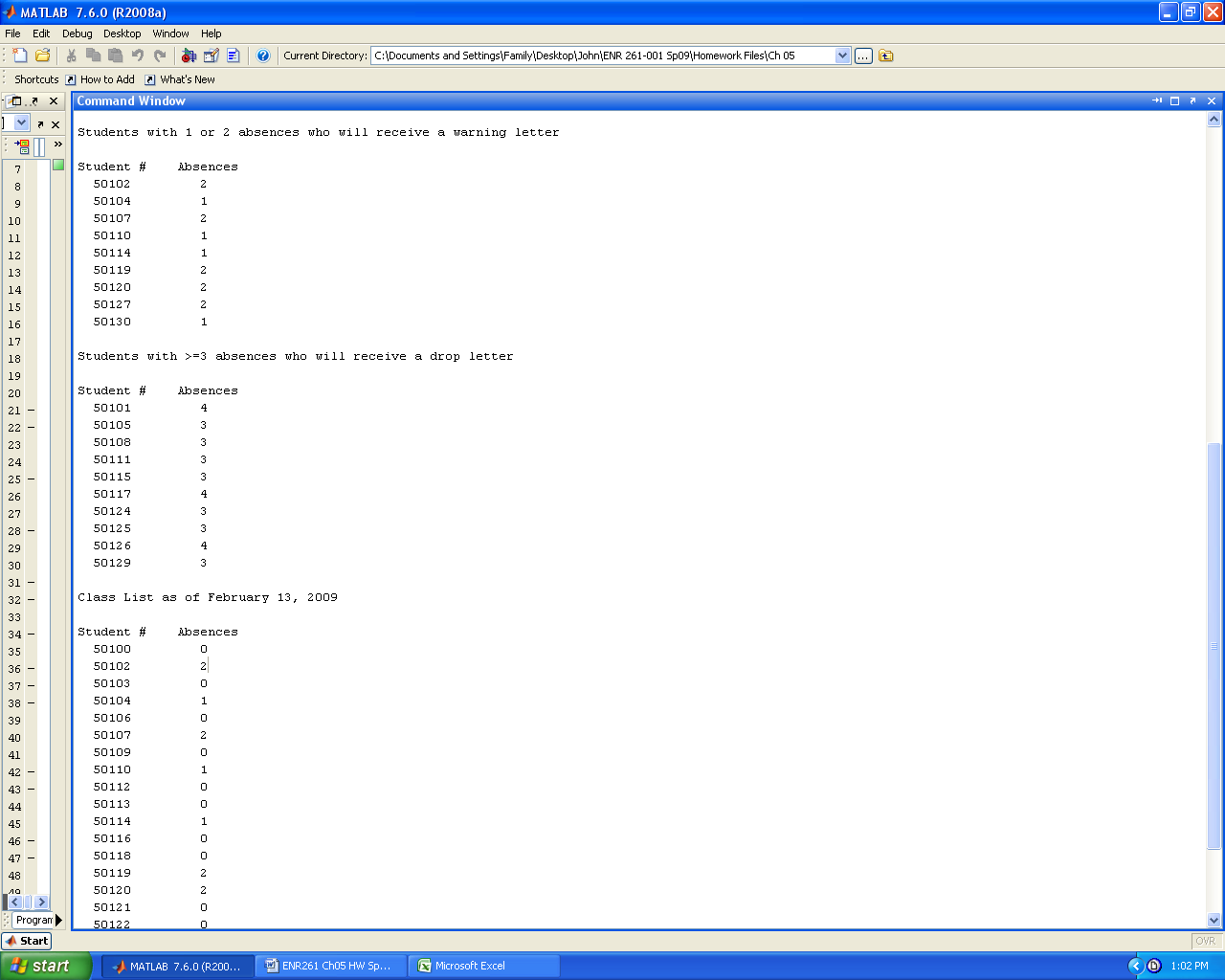
the ClassList.xlsx file.

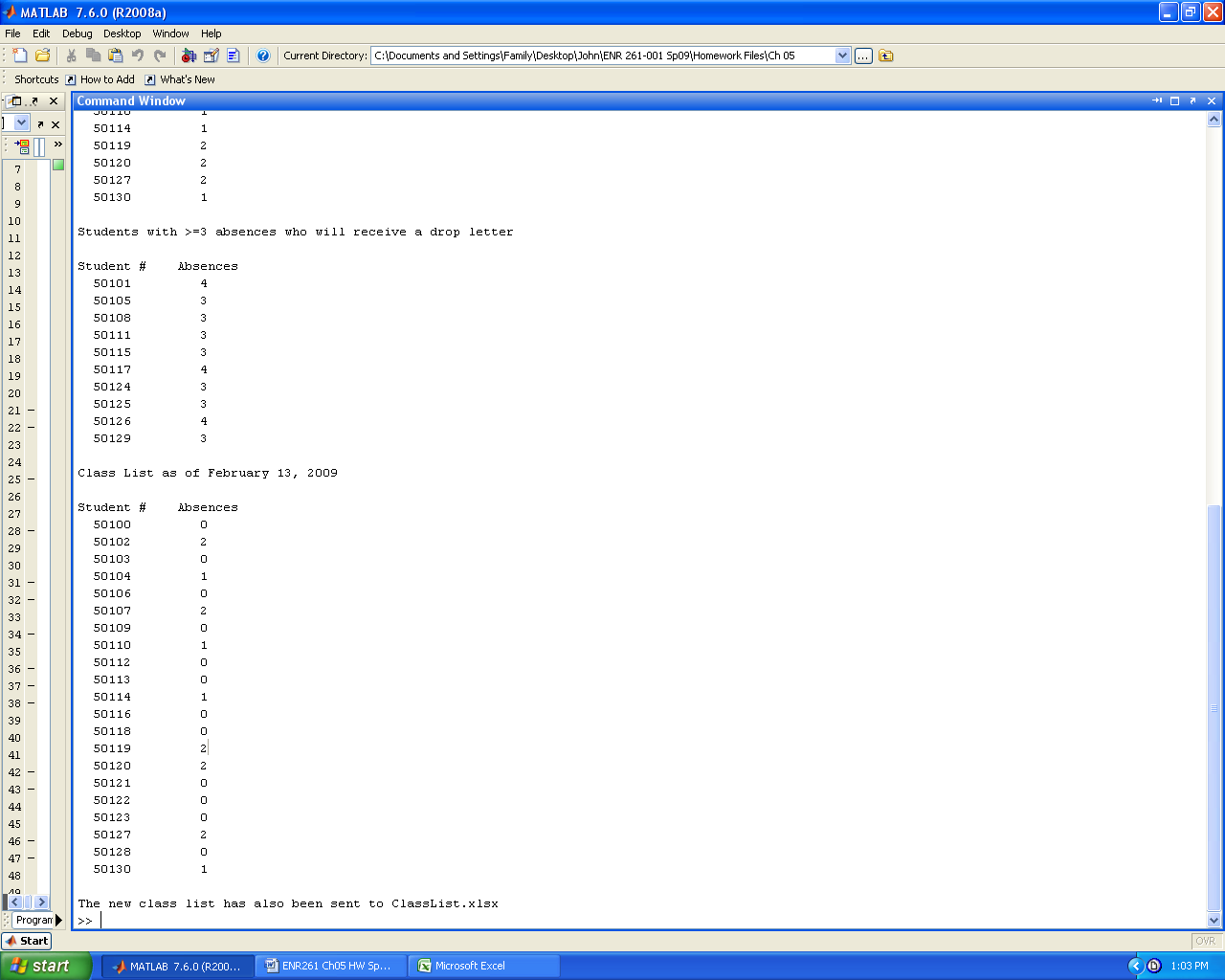
3. Print out your Excel sheets and the contents of your command window.

4. A sample of the command window is shown below.



**Continued on the Next Page**





Required File Name: **Program\_05\_2.m**

1. Perform the operations outlined below using Logical Vectors when appropriate.

Program Description:

This program creates 10,000 random resistors with a normal distribution

and mean of 100 ohms and standard deviation of 15 ohms.

Perform the following operations with the resistor data using logical

vectors where appropriate.

1. Output the total number of resistors created along with their actual

mean and standard deviation as they will differ slightly from the

values given.

2. Output the percentage of resistor with values > mean and < mean

3. Output the percentage of resistors within +/- 1 Standard Deviation

of the mean.

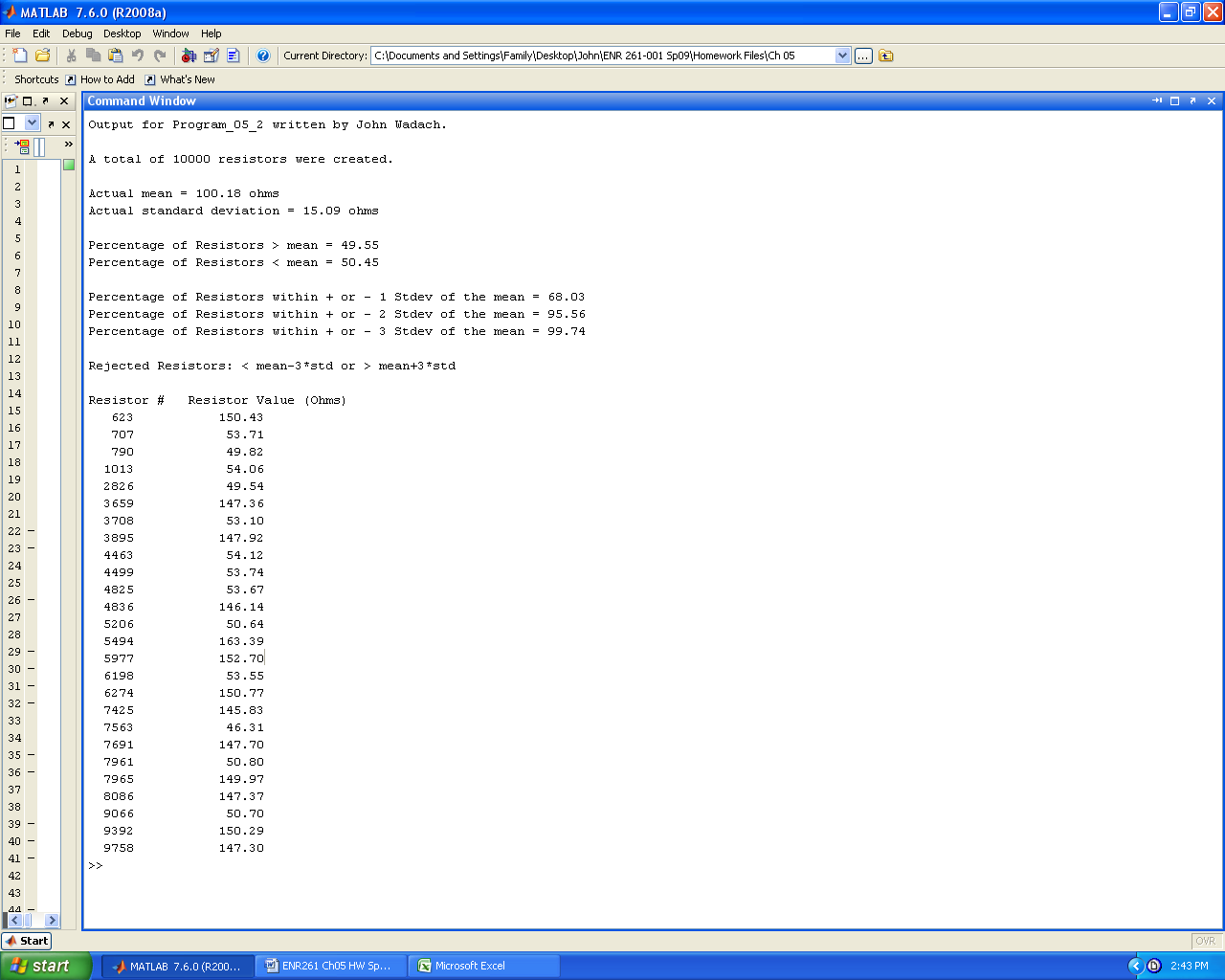
4. Repeat step 3 for +/- 2 Standard Deviations and +/- 3 Standard

Deviations.

5. Output the resistor position numbers and the corresponding resistor

values for all resistors outside of +/- 3 standard deviations of the mean

2. A sample of the output is shown below. Keep in mind that the results will vary for each run of your code.



Required File Name: **Program\_05\_3.m**

1. Perform the operations outlined below using Logical Vectors when appropriate.

Program Description:

This program creates 1,000 random resistors in batch 1 with a normal distribution and mean of 100 ohms and standard deviation of 15 ohms.

Perform the following operations with the resistor data using logical

vectors where appropriate

1. Output the total number of resistors created in batch 1 along with

their actual mean and standard deviation as they will differ

slightly from the values given.

2. Find the position number and corresponding resistance value for all

resistors in the batch that fall above the upper bound of mean + 1

std or below the lower bound of mean - 1 std. Hint: Use the find()

function, see the previous tutorials.

3. Output the number of resistors rejected and the batch number.

4. Output the position number and corresponding resistance value for all

resistors rejected in the batch.

5. If a batch has more than 1 rejected resistor all rejected resistors

must be recreated using the randn function and used to replace the

rejected resistor in the original vector of resistors.

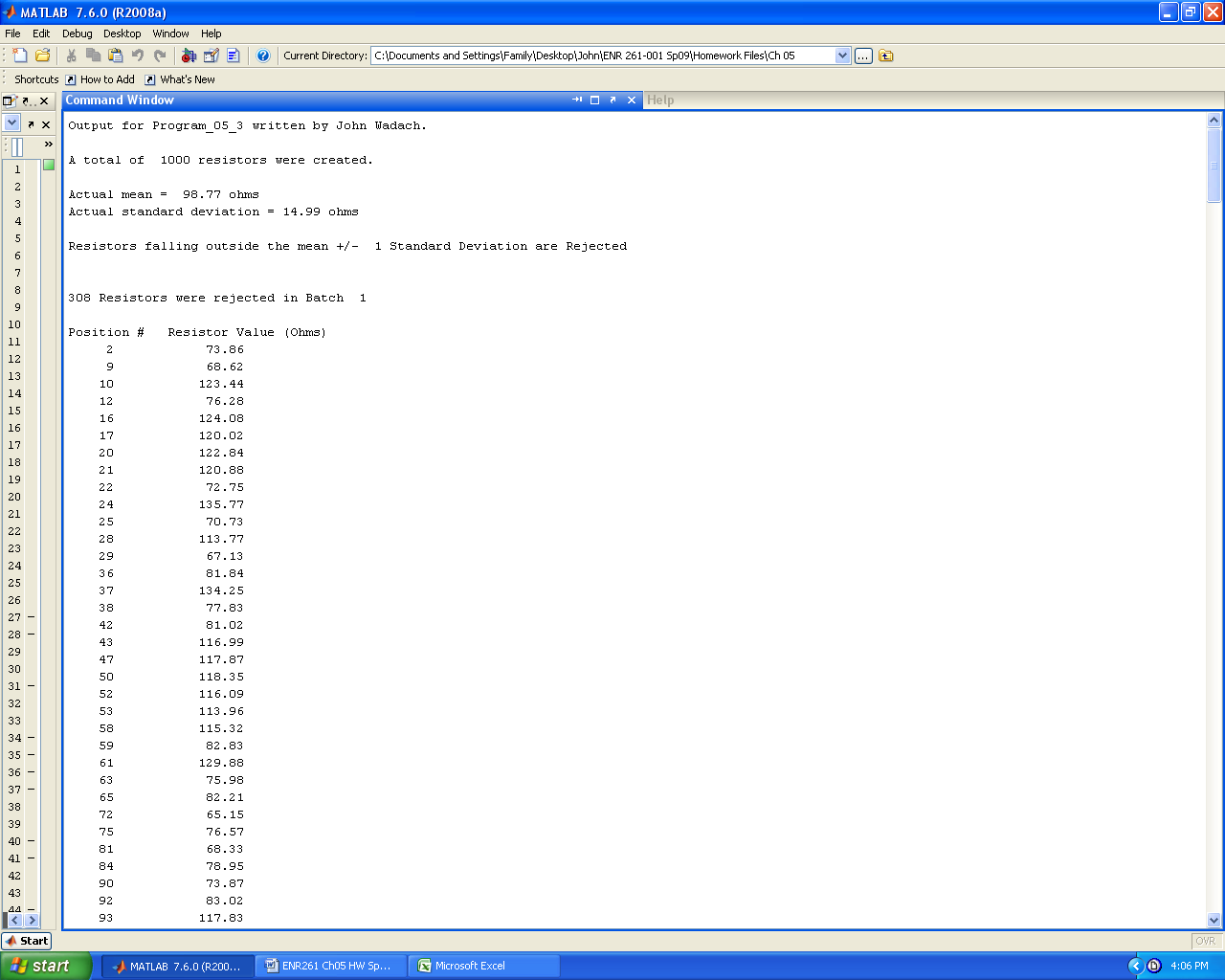
6. Repeat steps 2-5 until there are no resistors outside the lower and

upper bounds. Hint: Use a while loop (see tutorial\_03\_2) with the

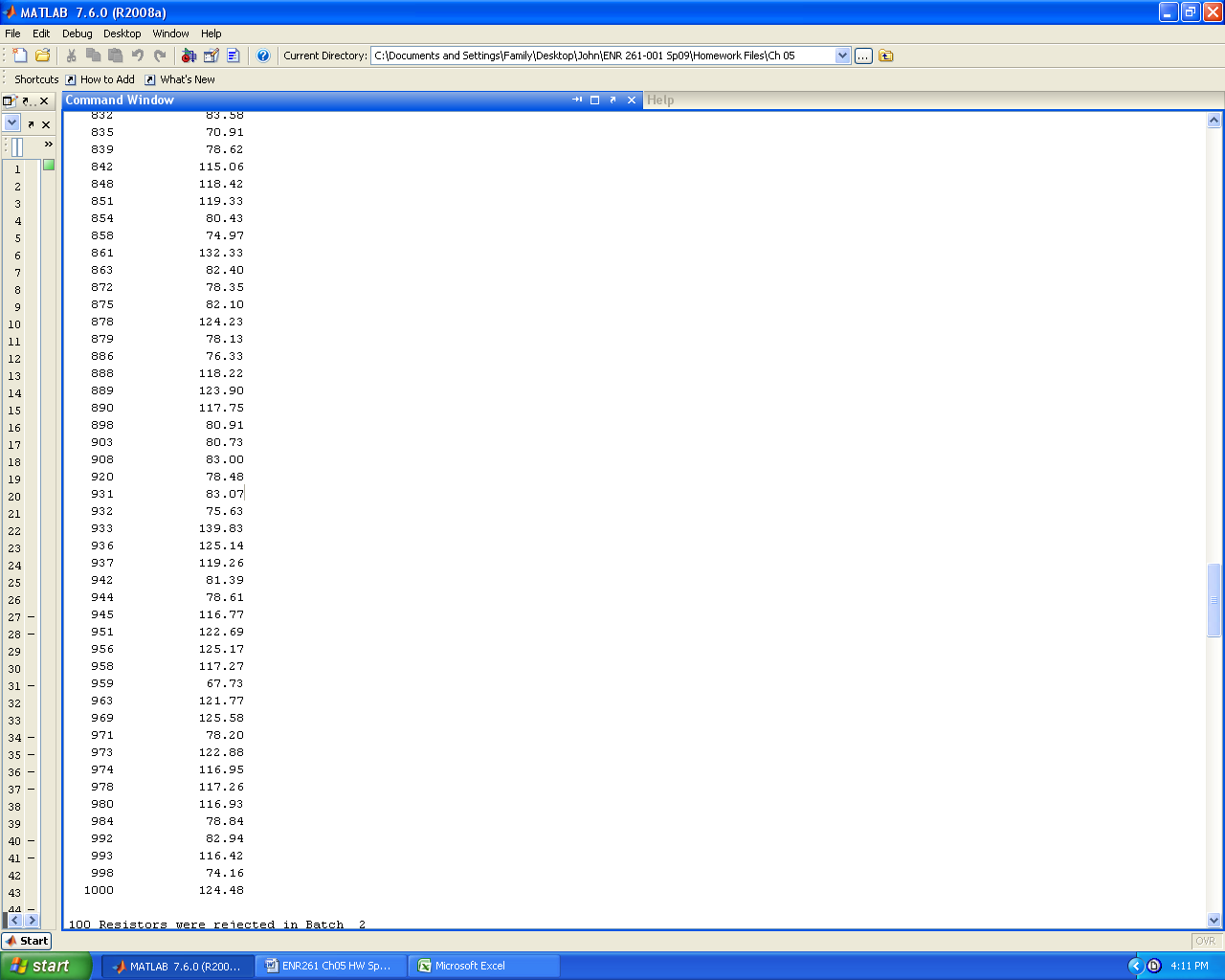
isempty() function (see page 133 of the text) to determine when to

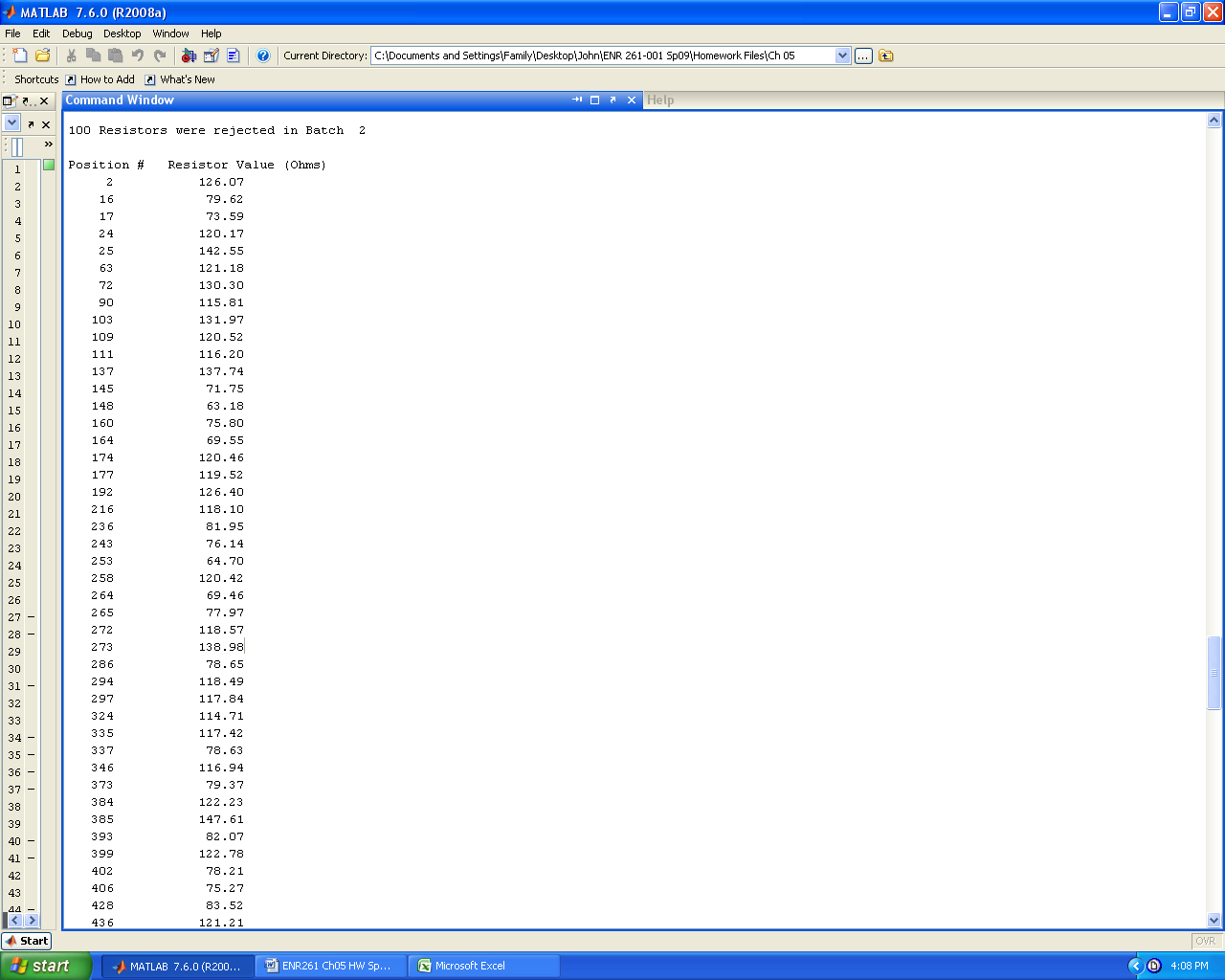
stop running new batches.

2. A sample of the output is shown below. Keep in mind that the results will vary for each run of your code. You can use the fact that about 1/3 of the values of each batch will be rejected as a way to determine if your output is reasonable.



Break in table as there were 308 rows in this table





Break in the table as there were 100 rows in this table

