Prepare complete solutions for each of the problems. Clearly present solutions in Word or on these pages as instructed. Each solution should be on a separate page with your name and the problem number at the top of each page. Your solutions must be well organized including clear labeling of parts, and documentation of all MATLAB commands and results. This exam consists of six questions on five pages.



1. Function with multiple I/O and comments (23 points): Consider the cone shaped solid shown in Figure 1. In this figure the height (h ), base radius (r ), and surface length (l ) are labeled. Based on these dimensions the surface length and surface area of a cone can be calculated using the following formula:

Surface length: $l= \sqrt{r^{2}+h^{2}}$

Surface area: $A= π\left(r^{2}+rl\right)$

Use the steps below to develop a function that calculates and outputs the surface length and surface area of a cone, given the base radius (r ) and the height (h ). All input and output must happen at the command line only. Program code and the setup table (part a) must be consistent.

**Figure 1**: A cone with a height (h ), base radius (r ), and surface length (l ) labeled. From: M.R. Spiegel, *Mathematical Handbook*, Schaum’s Outlines Series, McGraw Hill, 1968.

The function must be able to correctly handle vectors as the inputs.

a. Fill in the following information: **Parts a and b should be done on this page.**

|  |  |
| --- | --- |
| **Problem Goal (brief):** |  |
| **Inputs:** | **Variable Name** | **Description** | **Units/ Dimensions** | **Input Source\*** |
|  |  |  | Command line |
|  |  |  | Command line |
| **Outputs:** | **Variable Name** | **Description** | **Units/ Dimensions** | **Output type\*** |
|  |  |  | Command line |
|  |  |  | Command line |

1. Complete a hand calculation on a test case to compare to your program results.
Use r = 1.2 ft and h = 3 ft. as a test case.

**Parts c, d and e should be copied to a document and printed out together.**

c. Code with Comments: Copy your developed code to a word document. Use full comments on this program (including Problem ID, User introduction/help, Variable definition, Program Steps)

d. Scalar Execution: Run your developed code on the above test case. Copy the Command window call and result to the document with the code. Note: all I/O must be at the command line)

e. Vector Execution: Test the program with an input of two equal length vectors (one for the base radius and one for cone height). Add a copy the Command window call and result to your document.

1. Files & Arrays (10 points): When steel is stretched beyond its yield point it will undergo inelastic deformation and be permanently stretched. A tension test is conducted on a steel bar where the tension on the bar is slowly increased to past its yield point. The tension is then slowly decreased until there is no remaining tension on the bar. The results of this test are in the ASCII file ‘tension.csv’. In this file:
* Column 1 is the elongation of the bar in thousands of an inch,
* Column 2 is the tension corresponding to each elongation as the tension is increased,
* Column 3 is the tension corresponding to each elongation as the tension is reduced.

Both tension variables have units of pounds-force.

Prepare a script that will:

1. Input the File Name: The name of the data file must be specified (inputted) at the command line when the function is called (i.e., the name should not appear in the code).
2. Import the ASCII (.csv) file: Uses the input variable containing the file name to input the file into MATLAB
3. Displays the first five rows of data in the command Window.

Prepare a printout on a new page that includes: (1) the script,
 (2) the resulting command window call & output,

Script comments can & should be limited to: problem identification, student name, and
 identification of the program parts (i.e., parts a – c above).

1. Graphing (11 pts): Graph the two series for tension vs. elongation from problem 2.
	1. One series should be for increasing tension and the other for decreasing tension.
	2. Put both on one plot (i.e., one set of x-y axes).
	3. Plot the elongation on the x-axis (the two tensions on the y).
	4. Plot the entire data set (not simply the five rows displayed previously).
	5. Completely label the resulting graph and its two curves (including axis labels and legend).
	6. Make sure graph and fonts are easily read and follow standard graph format conventions.

Copy the commands used (either command window code or as a script) and the requested graph.

**If stuck** on successfully reading in the csv file: You can use the data file tension1.mat (with a loss of points for reading the csv file). At the command line type >> ***load tension1.mat*** to load the three variables elongation, TensionUp, and TensionDown. The file tension1.mat must be in your current working directory.

1. Structured Programming – loop (21 points): Figure 2 shows a MATLAB function that converts a negative number to -1, a positive number to +1, and zero to zero. Enter this code, and save it to your MATLAB test folder. You do not have to copy the comments.

function y=sgn(x)

% Determines the sign of

% an input variable.

% y = sgn(x)

% input: x = any real number

% output: y = -1 for x < 0

% 1 for x > 0

% 0 for x = 0.

if x<0

 y=-1;

elseif x>0

 y=1;

else

 y=0;

end

1. Validate this function as is and copy the validation runs to a document. On the document with the validation runs, briefly explain your validation logic (i.e., how the test cases you choose test that the function is working correctly in all aspects.). You do not need to include a copy of the code at this stage.
2. Modify this function so that it will handle vector inputs and evaluate them element-by-element. ***Comments may be limited to your name and problem Identification***. Add a copy the modified code to the document with the part a results.

**Figure 2:** MATLAB function for Problem 3 that flags the sign of a number.

1. Execute the modified function for the following two test cases:
2. x1= [4, - 2.1, 0, 3.6];
3. x2= [-1, 4, -5];

Add the results of these test cases the material from parts (a) and (b). Print out.

1. Structured Programming – Conditionals (21 points): A Microchip model MCP3204-CI/P integrated circuit (IC) is used to convert analog signals to digital signals. The price of these IC chips varies with the number purchased and is shown in the table below.

|  |  |
| --- | --- |
| Quantity Purchased | Price/IC |
| 1 – 34 | $ 4.15 |
| 35 – 149 | $ 3.23 |
| 150 – 599 | $ 2.35 |
| > 599 | Call for quote |

For example if the order is 50 chips select the $3.23 price and multiply by the number of chips: 3.19\*50 = $159.50. So the program should return $159.50.
N.B. the MATLAB command >> ***format bank*** will result in numbers formatted with two decimal places.

If the number of chips is more than 600 the program should return NaN as the command line output and display a text message “Call for Quote” in the command window.

Table 1 outlines the basic setup for this problem. The I/O in your program must match this setup.

**Table 1:** Goal and I/O structure for the Chip Price Problem

|  |  |
| --- | --- |
| **Goal**  | This program is to return the cost of a requested number of circuits, including price variation for quantity.  |
| **Inputs** | **Variable Name** | **Description** | **Units/ Dimensions** | **Input Source\*** |
| qty | The number of items to be purchased | Count | Command line |
| **Outputs** | **Variable Name** | **Description** | **Units/ Dimensions** | **Output type\*** |
| Cost | Total cost of the purchase or NaN | $  | Command line |
|  | Display ‘Call for quote’ for large qty. | text | Display  |

a. In the space below show hand calculations on an appropriate number of test cases to compare to your program results. Plan enough cases to provide for adequate program validation.

1. Develop your code. Comments may be limited to ID., student name and date.
2. Execute your code for the test cases calculated in part a.

Copy the function code developed and the execution for the test cases to a document and print out.

1. Empirical Models (14 points) Each of the two problems below shows a different set of data plotted three ways. From the plots deduce the likely model that will fit this data.

Your answer for each data set should include

* 1. The name of the likely model linear, exponential or power (not the name of the plot).
	2. The corresponding equation (using m and b for the model parameters) for that model. You are not expected to determine the values of m or b.
	3. A brief explanation of why you choose that type of model.

**Write answers on this page in the space provided after each data set’s graphs.**

* 1. First Data Set



* 1. Second Data Set

