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):**

Calculate the Surface length and Surface Area of a cone, given the base radius and height of the cone.

**Inputs: (**full name, variables to be used, units)

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Description | Units or Values | Input Source\* |
| r | Cone Base radius | length | Command line |
| h | Height of Cone | length | Command line |

**Outputs:**  (full name, variables to be used, units)

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Description | Units or Values | Output type\* |
| L | Surface length  | Length | Command line |
| SA | Surface area of cone | Length2 | 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.

Required case: r = 1.2 ft., h = 3 ft.

Surface length: $l= \sqrt{1.2^{2}+3^{2}}= \sqrt{10.44}=3.231$ ft

Surface area: $A= π\left(1.2^{2}+(1.2)(3.231)\right)=5.32π= $16.70 ft2

 Other test cases: r = 1 ft., h =2 ft.

Surface length: $l= \sqrt{1^{2}+2^{2}}= \sqrt{5}=2.24$ ft
 Surface area: $A= π\left(1^{2}+(1)(2.24)\right)=3.24π= $10.17 ft2

r = 1.8 ft., h = 4 ft.
Surface length: $l= \sqrt{1.8^{2}+4^{2}}= \sqrt{19.24}=4.39$ ft
Surface area: $A= π\left(1.8^{2}+(1.8)(4.39)\right)=11.14π= $34.98 ft2

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

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

function [A,L] = cone(r, h)

% S. Scott Moor December 2018

% This function return the surface area and surface length (distance

% from the base circumference to the tip for a right circular cone), given

% the base radius and the height of the cone. Program can handle vectors

% All measurements must use the same length measure

%

% Call: [A,L] = cone(r, h)

% Input Varables: r = radius of the cone base (length units)

% h = height of the cone (same length units)

% Output Variables: A = area (length units squared)

% L = edge length (length units)

% Calculate the edge length

L = sqrt(r.^2 + h.^2);

% Calculate the surface area

A = pi\*(r.^2 + r.\*L);

1. Scalar Execution: Run your developed code on the above test case. Copy the Command window call and result to the document with the code.

>> [A, L] = cone(1.2, 3)

A =

 16.7048

L =

 3.2311

Matches test calculation on previous page.

1. Vector Execution: Test the program with a range of values in a vector. Add a copy the Command window call and result to your document.

>> r = [1, 1.2, 1.8];

>> h = [2, 3, 4];

>> [A, L] = cone(r, h)

A =

 10.1664 16.7048 34.9829

L =

 2.2361 3.2311 4.3863

Works for vectors! (matches earlier calcs.)

1. Files, Arrays & Plotting (21 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 function that will:

1. Input the File Name: The name of the file to be read should be specified (inputted) at the command line when the function is called.
2. Loads 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.
4. Graphs the two series (i.e., two sets of data) for tension vs. elongation.
	* One series should be for increasing tension and the other for decreasing tension.
	* Put both on one plot (i.e., one set of xy axes).
	* Plot the elongation on the x-axis.
	* Plot the entire data set (not simply the five rows displayed).
	* Completely label the resulting graph and its two curves (including axis labels and legend).
	* Make sure graph and fonts are easily read and follow standard graph format conventions.

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

Prepare a print out that includes: (1) the script,
 (2) the resulting command window output, and
 (3) the requested graph.

**Function:**  function steel(fname)

% this program plots steel tension data read in from a file

% S. Scott Moor – Solution Only basic ID comments required

% Steps: a) Input file name from the command line (fname above)

% b) Loads the ASCII (.csv) file with the inputted filename into MATLAB.

stuff = importdata(fname); % for MATLAB

% stuff.data = csvread(fname)(2:end,:); % for Octave

x = stuff.data(:,1);

y1 = stuff.data(:,2);

y2 = stuff.data(:,3);

% c) Displays the first five rows of data in the Command Window.

disp('below are the first five rows of steel.csv')

disp(stuff.data(1:5,:))

% d) Graphs the two tension vs. elongation curves on one plot.

plot(x,y1,'p',x,y2,'s')

xlabel('Elongation (Thousands of an inch)')

ylabel('Tension (lb-force)')

legend('Increasing tension', 'Decreasing tension')

grid on

**Execution** >> steel('tension.csv')

below are the first five rows of steel.csv

 0 0 NaN

 1 11800 0

 2 21300 10100

 3 31100 20300

 4 38800 29700



1. Structured Programming (loop): 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. Comments may be limited to problem Identification.

function y=sgn(x)

% Determines the sign of an input variable.

% y = sgn(x)

% input: x = any real number.

% output: y = Returns: -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

(a) Validate this function as is and copy to a document. Briefly explain your validation logic (i.e., how this proves the function is working).

>> sgn(5)

ans =

 1

>> sgn(0)

ans =

 0

>> sgn(-2.3)

ans =

 -1

Each branch of the conditional is tried. The program produces the expected results of a 1 for positive 5, a 0 for 0, and a -1 for negative 2.3.

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

(b) Modify this function so that it will handle vector inputs and execute it for these cases:

1. x1= [4, -2.1, 0, 3.6];
2. x2= [-1, 4, -5];

Show your modified code with id comments, and the execution

Code:

function y=sgn(x)

**Execution**

>> x1 = [4, -2.1, 0, 3.6];

>> x2=[-1, 4, -5];

>> sgn1(x1)

ans =

 1 -1 0 1

>> sgn1(x2)

ans =

 -1 1 -1

% Determines the sign of an input variable.

% Modified by S. Moor to handle vector input

% April 2015

%

% y = sgn(x)

% input: x = any real number.

% output: y = Returns: -1 for x < 0

% 1 for x > 0

% 0 for x = 0.

% preallocate vector size (optional)

y = zeros(size(x));

% for loop used for element-by-element calculation

for k = 1:length(x)

 if x(k) < 0

 y (k) = -1;

 elseif x(k) > 0

 y(k) = 1;

 else

 y(k) = 0;

 end

end

1. Structured Programming (Conditionals): 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: 3.23\*50 = $161.50. So the program should return $161.50. 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 for the command line output and display a text message “Call for Quote”. 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.

 **c. Execution**

>> format compact

>> format bank

>> chips(15)

ans =

 62.25

>> chips(44)

ans =

 142.12

>> chips(200)

ans =

 470.00

>> chips(600)

Call for Quote

ans =

 NaN

🡺matches expectation

|  |  |  |
| --- | --- | --- |
| Quantity Purchased | Price/IC | Total (Price x qty) |
| 15 | $ 4.15 | $ 62.25 |
| 44 | $ 3.23 | $142.12 |
| 200 | $ 2.35 | $470.00 |
| 620 | Call for quote | NA |

1. Develop your code. Comments may be limited …
2. Execute your code for the test cases calculated in part a.

function Cost = chips(qty)

% Chip (qty) calculates and returns the total cost for an order of ICs

% Lab Practical ENGR 128 # 4. Dec. 2018 S. Scott Moor

if (qty > 0) && (qty <35)

 Cost = qty \* 4.15;

 elseif qty < 150

 Cost = qty\*3.23;

 elseif qty < 600

 Cost = qty\* 2.35;

 else

 Cost = NaN;

 disp('Call for Quote')

end

1. (8 pts.) 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. Please name the model, give its equation (using m and b for the model parameters) and briefly explain why you choose that type of model.
	1. First Data Set



Name: exponential

Equation: y = bemx

Explanation: The relatively straight arrangement of the data on the semilog plot indicates that this is the likely model

* 1. Second Data Set



Likely Model

Name: power law model.

Equation: y = bxm

Explanation: The relatively straight arrangement of the data in the log-log plot suggests the power model is likely.