

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Department of Mechanical Engineering

ME123

Computer Applications I

EXAM 2 – WRITTEN PORTION

NAME _____

SECTION NUMBER _____

CAMPUS MAILBOX NUMBER _____

EMAIL ADDRESS _____@rose-hulman.edu

Written portion	/40
Computer portion	/60
Total	/100

Problem 1:

Consider the code scrap shown below:

```
clc
clear variables

t(1,1) = 0;
n = 1;
dt = 0.2;

while t(n,1) < 0.5
    t(n+1,1) = t(n,1) + dt;
    n = n+1;
end
```

After execution of this code,

- What is the value of **n**? _____
- What is the value of the final element in the **t** array? _____

Problem 2:

For each type of problem shown below, we could use either a **while...end** loop or a **for...end** loop.

Circle the type of repetitive structure that would be the *most* appropriate for the type of problem.

- (circle one) **while for** Countdown from 10 to 0 to initiate a space shuttle launch.
- (circle one) **while for** Calculate a new estimated value of a function until an error estimate is small enough.
- (circle one) **while for** Keep checking for input validity until the input is within a valid input range.
- (circle one) **while for** Record the voltage for 25 sensors, one at a time.

Problem 3:

A MATLAB program called **Main_program** and a user-defined function called **plus_times** are shown.

<pre>% Main_program % Inputs x = 5; y = 2; % Calculations [m,n] = plus_times(x,y) % Outputs fprintf('x = %2.1f \n', x) fprintf('y = %2.1f \n', y) fprintf('m = %2.1f \n', m) fprintf('n = %2.1f \n', n)</pre>	<pre>function [x,y] = plus_times(a,b) % User-defined function that computes % the sum of the input arguments and % the product of the input arguments x = a + b; y = a * b;</pre>
--	--

What does the main program print for x , y , m , and n ? (Fill in the blanks.)

$x =$ _____

$y =$ _____

$m =$ _____

$n =$ _____

Problem 4:

If we fill a tank that has a leak, the differential equation is

$$\frac{dh}{dt} = -v - k\sqrt{h}$$

where h is the height of the fluid in the tank, t is time, v is the fill rate, and k is a discharge coefficient. (v and k are constants.) Write the appropriate mathematical (not MATLAB) expression for the Euler's method approach to this problem.

$$h_{n+1} = \text{_____}$$

Problem 5:

Recall that the quadratic formula solves for the roots of a quadratic equation (second-order polynomial)

using the formula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$. We expect two answers: x_1 and x_2 . Given this information, the

following user-defined function has been written:

```
% quadratic.m
% This function computes the roots of a quadratic equation
% of the form (a x^2 + b x + c).
%
% Inputs are a,b,c which are the coefficients of the powers of x.
% Outputs are x1,x2 which are the roots of the equation
%
x1=(-b-sqrt(b^2-4*a*c))/(2*a);
x2=(-b+sqrt(b^2-4*a*c))/(2*a);
```

- a. Based on the function comments and code, write first line of this function (line 01, the function definition statement).

- b. Write the line of MATLAB code to call this function to solve the quadratic equation:

$$px^2 + qx + r$$

given that the variables p , q , and r are already assigned values in the main program and that you wanted to store the roots using variable names **root1** and **root2**.

Problem 6:

Consider the thermocouple problem

$$\frac{dT}{dt} = -k(T - T_a)$$

where T is temperature, t is time, and T_a and k are constants.

(a) Fill in the blanks in the following code to create a correct MATLAB implementation of the Euler solution to the problem. Do not change any of the existing code:

```
dt=0.001;
k=1.5;
Ta=200;
tmax=5;

n=_____
T(1,1)=0;
t(1,1)=0;
while t(n,1) < tmax;
    t(n+1,1)=t(n,1)+dt;

    T(n+1,1)=_____
    n=n+1;
end
plot(t,T)
```

(b) Now fill in the blanks in this (different but related) code to create a correct MATLAB implementation of the Euler solution to the problem. Do not change any of the existing code:

```
dt=0.001;
k=1.5;
Ta=200;
tmax=5;

n=_____
T(1,1)=0;
t(1,1)=0;
while t(n-1,1) < tmax;

    t(n,1)=_____

    T(n,1)=_____

    n=n+1;
end
plot(t,T)
```

Problem 7

We wish to write a function for an exact solution of the form

$$x = x_0 - \frac{1}{2}gt^2$$

for times between 0 and t_{max} . Fill in the blanks in this code to create a correct MATLAB function for this problem. Do not change any of the existing code.

```
function [x,t]=Exact(dt,x0,tmax);  
g=32.2;  
n=1;  
  
x(1,1)=_____  
t(1,1)=0;  
while t(n,1) < tmax;  
    t(n+1,1)=t(n,1)+dt;  
  
    x(n+1,1)=x(_____,1)-0.5*g*t(_____,1)^2;  
    n=n+1;  
end
```

EXAM 2 – COMPUTER PORTION**Instructions:**

For this portion of the exam you will create two files:

- a main program m-file called `lastname_firstname.m`
- a function m-file called `E_calc_lastname.m`

Before beginning the problems, open and save both m-files now.

Main program

Include your name, section number, and CM number in the header section of your code. There should be no output other than what is asked for.

Start problem 1 in this file with the lines:

```
%% Problem 1
clc
clear variables
close all
```

Start problem 2 in this file with the line (clear only the variables):

```
%% Problem 2
clear variables
```

Function

Include your name, section number, and CM number in the header section of your function. There should be no output other than what is asked for.

Problem #1 (20 pts)

Recall that, in MATLAB, we can define an array A with the following command:

```
A = [2; -5; -3; 6; 8; 9; 2; -7; 1; -9];
```

- Type the array A into your main program after the comment line `%% Problem 1`.
- Write a program that counts the number of negative numbers in this array.
- Display this count (a single integer answer) on screen.

Problem #2 (40 pts)

- Download the EXCEL file named `curve_fit.xls` from the course web page at <http://www.rose-hulman.edu/ME123/courseware.shtml>.
- Save the EXCEL file in the same directory as your Exam 2 m-files.
- Begin this section of code after the comment line `%% Problem 2.`

The EXCEL spreadsheet contains two columns of data recorded by a pressure transducer during a calibration procedure. The first column is the voltage (in mV) and the second column is the corresponding pressure (in kPa).

(a) Start this section of your code with the comment line: `% Part (a).`

- Read in the data and plot pressure (y axis) as a function of voltage (x axis).
- Use descriptive axis labels and a descriptive title for your plot.
- Plot these data using *data symbols with no connecting lines*.

(b) Start this section of your code with the comment line: `% Part (b).`

Your supervisor believes that the relationship between pressure and voltage is linear, that is,

$$P = aV$$

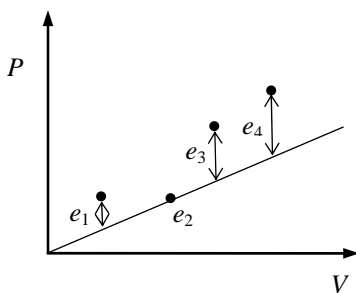
- Create a line of this form, with $a = 1.2$, and add it to your data plot from part (a).
- Plot this line as a solid line with no additional data symbols.
- *If you cannot get this portion to work correctly, we suggest that you skip to part (c).*

(c) Start this section of your code with the comment line: `% Part (c).`

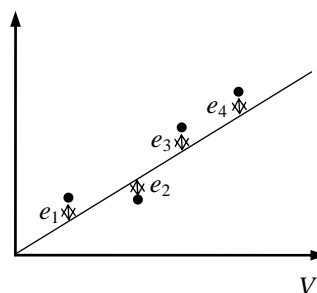
The total error (E) between the line and the data can be written as

$$E = \sum_i e_i^2 = \sum_i (P_i - aV_i)^2$$

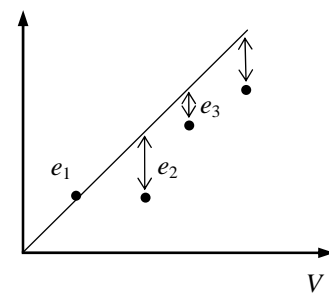
where $i = 1 \dots n$ and n is the total number of data points. If the slope a is too small (Case 1 in the figures), the error is large. If a is too large (Case 3) the error is also large. If a is just right (Case 2), the total error will be small.



Case 1: a is too small



Case 2: a is about right



Case 3: a is too large

Write the function called `E_calc_lastname.m` that

- accepts as inputs: proposed slope a , the pressure array P , and the voltage array V ,
- uses a and the elements of the P and V arrays to compute the total error E given by

$$E = \sum_i e_i^2 = \sum_i (P_i - aV_i)^2$$

- returns the value of E to the main program.

Test your function from the main program by

- Setting $a = 0.5, 1.2,$ and 2.0
- Print the result E to the screen with appropriate words using an `fprintf` statement, for example:

```
a=0.50      E= X.XXXXXe+XXX
a=1.20      E= X.XXXXXe+XXX
a=2.00      E= X.XXXXXe+XXX
```