ENGR 130 Midterm Exam 10/10/23

Note the following:

- This exam is closed book, closed notes.
- You may not use a calculator or any additional paper. If you need to write notes, use the margins of this paper.
- Please turn off your cell phone and put it inside your bag.
- You have 75 minutes to complete this exam. When you are finished, you may turn your exam in and leave.
- This exam consists of 9 problems on 7 pages, totaling 100 points.
- Partial credit will be given for partially correct work. Write neatly and clearly – if we can't read it or understand it, we can't give you credit for what you know.

Section 1: Short Answer

PEMDAS

1) (6 pts) Write the output of each snippet of code in the blank provided below it. If a command will result in an error, write "error".

a) `7 - 5 * 5 ^ 0 + 2`

$7 - 5 \times 1 + 2$

$7 - 5 + 2$

$2 + 2$

ans = 4 ✓

b) `4 = houses`

error ✓

c) `matrix = [5 25 14; 27 13 11];`

`min(min(matrix))`

ans = 5 ✓

6

2) (15 pts) In each blank below, write one line of code to perform the operations described. No points will be awarded for manually defining vectors by typing in the values.

a) Calculate a variable `A`, equal to πr^2. You may assume that the variable `r` needed for the calculation already holds a value.

A = pi*r^2 ✓

b) Create the vector `B`, with the values 20, 15, 10, 5, 0, -5, and -10, in that order, in a way that requires less than 15 characters, not including spaces.

B = 20:-5:-10; ✓

c) Load all the variables within the file `states.mat`.

load('states.mat') ✓

d) There is a row vector called `years`, containing a list of years that each USA state was founded. Create a variable called `recent` that contains the numbers in `years` that are equal to or greater than 1850.

recent = years(years>=1850); ✓

e) Save the variable `recent` to a file called `new_file.mat`.

save('new_file.mat','recent') ✓

+15

3) (6 pts) Several variables are assigned values as shown below:

```
cars = 16;
trucks = 12;
buses = 5;
```

On each line below, write the output of the statement above.

a) `(cars < trucks + buses) & (trucks ~= buses)`

16 < 17 and 12 ≠ 5

1 1

1

b) `(trucks > cars) | (buses <= 6)`

12 > 16 or 5 ≤ 6

0 1

1

c) `(trucks < 11 | cars >= buses) & (trucks ~= cars - buses)`

12 < 11 or 16 ≥ 5 and 12 ≠ 11

1 1 1

1

+6

2

+21

4) (12 pts) Given two MATLAB matrices:

```
X = 21   36   15          Y = 14   30   10
     9   27   60                8   22   40
```

Write one line of MATLAB code to perform the following operations:

a) Subtract every element in Y from the corresponding element in X and store the result in Z.

Z = X - Y; +3

b) Multiply every element in Y by 2 and store the result in Q.

Q = Z * Y; +3

c) Cube every element in X and store the result in G.

G = X.^3 ; +3

d) Create a vector containing sixty numbers equally spaced between 0.35 and 43.68 and store the result in T.

T = linspace(0.35, 43.68, 60) ; +3

+12

Section 2: Analyzing & Writing Code

5) (15 pts) A vector called `nums` contains an unknown number of non-zero integers. Using a `for` loop, write a code segment that will iterate through the vector and sort the contents into two arrays based on the following conditions:

- All elements that are greater than or equal to 42 should be placed into a vector named `answers`.
- Any negative numbers should be placed into a vector named `subzeros`.

You may not need to use all the lines provided.

```
size_nums = length(nums);
ans_idx = 1;
subz_idx = 1;
for idx = 1:size_nums
    if nums(idx) >= 42
        answers(ans_idx) = nums(idx);
        ans_idx = ans_idx + 1;
    elseif nums(idx) < 0
        subzeros(subz_idx) = nums(idx);
        subz_idx = subz_idx + 1;
    end
end
```

15

6) (8 pts) Rewrite your code from Problem 5 using a `while` loop.

```
size_nums = length(nums);
idx = 1;
ans_idx=1;
subz_idx=1;
while idx <= size_nums
    if nums(idx) >= 42
        answers(ans_idx)=nums(idx);
        ans_idx= ans_idx+1;
    elseif nums(idx) < 0
        subzeros(subz_idx)=nums(idx);
        subz_idx=subz_idx+1;
    end
    idx=idx+1;
end
```

+8

7) (20 pts) A thermocouple is a device that measures temperature and outputs voltage. A thermocouple was calibrated by immersing the device into fluids of known temperatures and recording the output voltage (see data table below).

Temperature (°C)	Voltage (V)
0	0.0
260	0.2
550	0.4
900	0.7

Write a code snippet below to create the variables `Temp` and `Volts` and plot `Temp` vs. `Volts` as points with no connecting line. You may not need to use all the lines provided.

```
Temp = [0, 260, 550, 900];
Volts = [0, 0.2, 0.4, 0.7];
plot(Volts, Temp, '*')
% labels/title/legend at end of code
```

The code below calculates a line of best fit and adds that best fit line to the graph you created above. Fill in the missing lines to complete the code.

```
% Calculate fitting coefficients for a linear curve fit
fit_coef = polyfit(Volts, Temp, 1);

% Create vector of voltages to use for plotting curve
voltfit = linspace(min(Volts), max(Volts), 100);

% Create vector of fitted temperatures
fit_data = polyval(fit_coef, voltfit);

hold on;

% Graph fitted line
plot(voltfit, fit_data, 'b-')

% Add required text to graph
xlabel('Voltage (V)')
ylabel('Temperature (°C)')
title('Thermocouple Device Data of Known Liquids')
legend('Thermocouple points', 'fitted data')
hold off;
```

20

Section 3: Laboratory Questions

Dirty Water

8) (8 pts) For each statement below, circle the one word in the parentheses that best completes each sentence.

- A water sample with high turbidity contains a (lower **OR** higher) number of particles than a sample with low turbidity.
- A water sample with high turbidity would have a (lower **OR** higher) number of NTUs than a sample with low turbidity.
- A water sample with high turbidity allows (less **OR** more) light to pass through the sample than a sample with low turbidity.
- A water sample with high turbidity is likely (less **OR** more) safe to drink than a sample with low turbidity.

9) (10 pts) For each pair of breadboard holes below, circle "connected" if the holes are electrically connected or circle "not connected" if the breadboard holes are not electrically connected.

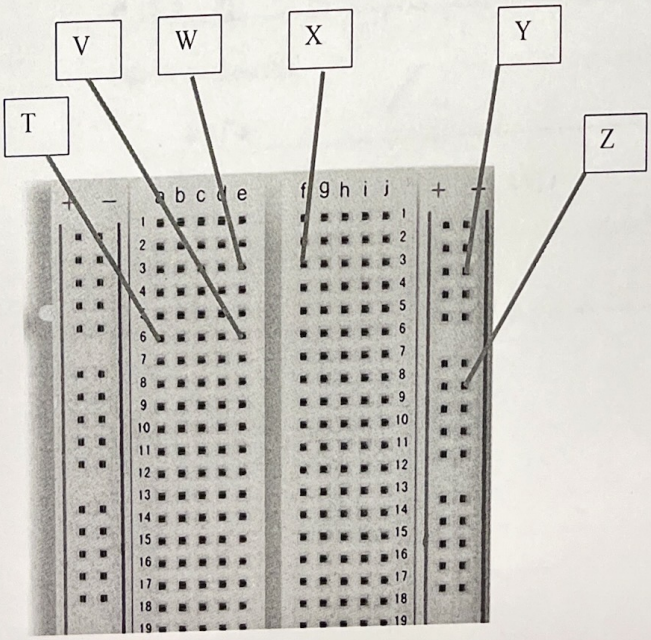
V & W : connected **not connected**

T & V : **connected** not connected

W & X: connected **not connected**

X & Y : connected **not connected**

Y & Z: **connected** not connected



18