SHORT TRACING AND SHORT ANSWERS (30 pts, recommended 10 mins)

1. Identify each of the following statements as true or false, and correct the false statements, or give a counterexample (3 pts each)
   1. Every if statement can be written as a switch statement.

T F

* 1. Though there can be an infinite while loop, there is no such thing as an infinite for loop.

T F

* 1. Every function that applies to vectors also works on cell arrays containing only doubles.

T F

1. What is stored in the variables newca1 and newca2 after the following code is run? If the code errors for a variable, write ERROR and briefly explain why there is an error to the right (3 pts each).

ca = {'blue' 'green' 'red'};

vec = [1 2 3];

newca1 = {ca vec};

newca2 = [ca; vec];

newca1: \_\_\_\_\_\_\_\_\_

newca2: \_\_\_\_\_\_\_\_\_

1. What is stored in the variable arr after the following code is run? If the code produces an error, write ERROR and explain why (3 pts).

arr = ones(4,4);

for i = 1:length(arr)

for j = 1:length(arr)

if i < j

arr(i,j) = 0;

end

end

end

arr: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. What is stored in the variable str after the following code is run? How many times does the while loop run?

str = 'to h\*\*\* with Georgia!';

i = 1;

while str(i)==lower(str(i))

if ~(str(i)>='a' && str(i)<='z')

str(end) = [];

else

str(i) = str(i)-'a' + 'A';

end

i = i + 1;

end

str: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

How many times did the loop run? \_\_\_\_\_\_\_\_

SHORT CODING (20 pts, recommended 15 mins)

1. Write 1-10 lines of code that does each of the following tasks. Do NOT write functions, simply write code blocks.
   1. Write code that prints the phrase 'OH NO!' onto fifteen consecutive lines of a text file called 'whoops.txt'. Do not include an extra line at the end of the text file (7 pts).

fh = fopen(‘whoops.txt’,’w’);

first = true;

newline = ‘\n’;

message = ‘OH NO!’;

for x = 1:15

if first

fprintf(fh,’%s’,message);

first = false;

else

fprintf(fh,’%s%s’,newline,message);

end

end

* 1. Write code that converts every value of class double inside of a one-dimensional cell array called ca to its corresponding string (i.e., the number 3 becomes the string '3'). Store the answer back in ca. (7 pts)

for x = 1:length(ca)

if strcmp(class(ca{x}),’double’)

ca{x} = num2str(ca{x});

end

end

* 1. Given a double called num, output the word 'Even' if the number is an even integer, 'Odd' if the number is an odd integer, and 'Neither' if the number isn't even or odd (for instance, if it's a decimal). Store your answer in a variable called ans. To receive full credit, you must use a switch statement (6 pts).

evenodd = mod(num,2);

switch evenodd

case 0

ans = ‘Even’;

switch 1

ans = ‘Odd’

otherwise

ans = ‘Neither’

end

1. Consider the following code and answer the questions that follow. (20 pts, recommended 10 mins)

1 function [numWords numLetters] = countThings(filename)

%This function calculates the total number of words in a text file, and %the total number of letters in each of these words.

2 fh = fopen(filename);

3 line = fgetl(fh);

4 numWords = 0;

5 numLetters = 0;

6 while ~isempty(line)

7 [word rest] = strtok(line);

8 while ~isempty(rest)

9 numWords = numWords + 1;

10 numLetters = numLetters + length(word);

11 [word rest] = strtok(rest);

12 end

13 line = fgetl(fh);

14 end

15 fclose(fh);

16 end

A file called 'file1.txt' contains the following text (the asterisks are NOT part of the text file).

This is a text file

Good times

Out of ideas

1. This code produces an infinite loop. Which line of code is at fault? Write the correct code to replace the line at fault (4 pts).
2. Assume that you've made the change in part a) and the code compiles and runs successfully. When you run the function using file1 as an input, however, the function outputs that there are 7 words, not 10. Explain why the code is not functioning as intended, and give one change that could be made to the code to make numWords function as intended. *Hint: Think about how many times the inner while loop runs.* (7 pts)
3. For each of the following scenarios, circle the choice that describes the effect the choice will have on the code (3 pts each).
4. fgets() is used instead of fgetl()
   1. The output of numWords will change.
   2. The output of numWords will not change.
5. Line 1 of the text file is replaced with the string 'This isn't a text file'.
   1. The apostrophe will be counted in numLetters.
   2. The apostrophe won't be counted in numLetters.
6. Line 3 of the text file is replaced with the string 'Out! of ideas'.
   1. The exclamation point will be counted in numLetters
   2. The exclamation point won't be counted in numLetters.
7. Write the following function. (30 pts, recommended 20 mins)

Function Name: gasPressure

Inputs (1): (char) The filename of an Excel file (will end in '.xls')

Outputs: None

Output Files (1): An edited Excel file

You've done a series of measurements for the pressure inside a closed cylinder for various gases, and stored your experimental results in an Excel file. However, you'd like to do a bit of data analysis before publishing the results.

Each row of the text file represents a single gas, and each column of the text file represents a numerical measurement of the pressure for various trials-- except for the first column, which contains the names of the gases. (See the example file formatted below). The file will always be formatted in exactly this way, though the number of gases and trials might vary.

To analyze your data, calculate the mean and standard deviation of your measurements for each gas. You may use the MATLAB built-in function std. Then, add a column to the end of the Excel file titled "Measured Range". For each gas in this column, store a string in the format 'MEAN ± STANDARD DEVIATION', where the MEAN and STANDARD DEVIATION are replaced by their actual values.

Write a new, edited Excel file, where the filename is identical to the original filename, but with the phrase \_edited\_ concatenated before the '.xls'.

Example:

The file 'measurements.xls' contains the following data.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Gas | Trial 1 | Trial 2 | Trial 3 | Trial 4 | Trial 5 |
| Oxygen | 1500 | 1485 | 1470 | 1490 | 1510 |
| Nitrogen | 1700 | 1685 | 1660 | 1660 | 1676 |
| Hydrogen | 850 | 860 | 860 | 860 | 855 |
| Helium | 1140 | 1120 | 1140 | 1120 | 1140 |

If you call gasPressure('measurements.xls')

It creates a new Excel file called 'measurements\_edited.xls' with the following data:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Gas | Trial 1 | Trial 2 | Trial 3 | Trial 4 | Trial 5 | Measured Range |
| Oxygen | 1500 | 1485 | 1470 | 1490 | 1510 | 1491 ± 1.5 |
| Nitrogen | 1700 | 1685 | 1660 | 1660 | 1676 | 1676 ± 1.7 |
| Hydrogen | 850 | 860 | 860 | 860 | 855 | 857 ± 4.5 |
| Helium | 1140 | 1120 | 1140 | 1120 | 1140 | 1132 ± 1.1 |

Helpful hints:

-Don't worry about rounding anything.

function deviation = std(vec)

% Outputs the standard deviation of a vector