Problem Set 08
Relational and Logical Operators

Instructions:

1. This problem set contains paired and individual programming problems. Each problem has a set of deliverables that needs to be submitted. You are responsible for following the appropriate guidelines and instructions below. Create appropriately-named files as instructed
2. Save all files to your Purdue career account in a folder specific to PS08.
3. Compress all deliverables into one zip folder named **PS08_yourlogin.zip**. Submit the zip file to the Blackboard drop box for PS08 before the due date.

Problem Set

Item	Type	Deliverable
Problem 1: Truth Table	Individual	PS08_truth_tables.docx
Problem 2: Indiana Corn Yield	Paired	PS08_yieldcompare_*yourlogin1*_*yourlogin2*.m PS08_yieldcompare_*yourlogin1*_*yourlogin2*_report.pdf Any data file loaded into your program
Problem 3: Thermocouple	Individual	PS08_thermocouple_*yourlogin*.m PS08_thermocouple_*yourlogin*_report.pdf Any data file loaded into your program

Truth Table Answer Sheet

You must use the truth table answer sheet template that is provided in the Assignment Files. Follow any additional instructions that appear in the answer sheet. Be sure to fill out the header information on the Answer Sheet

Problem 1: Logic Truth Tables

Individual Programming

Problem Setup

In logic, a true statement is assigned the value 1, and a false statement is assigned the value 0.

When writing and evaluating logical statements using truth tables, the following MATLAB operators are used as symbols:

Symbol	Logic Meaning
|	OR
&	AND
~	NOT

Problem Set 08
Relational and Logical Operators

For example, for the statement **A & ~xor(B,C),** the truth table would be:

	B=0		B=1	
	C=0	C=1	C=0	C=1
A=0	0 & ~xor(0,0) → 0 & ~0 → 0 & 1 → 0	0 & ~xor(0,1) 0 & ~1 → 0 & 0 → 0	0 & ~xor(1,0) → 0 & ~1 → 0 & 0 → 0	0 & ~xor(1,1) → 0 & ~0 → 0 & 1 → 0
A=1	1 & ~xor(0,0) → 1 & ~0 → 1 & 1 → 1	1 &~ xor(0,1) → 1 & ~1 → 1 & 0 → 0	1 & ~xor(1,0) → 1 & ~1 → 1 & 0 → 0	1 & ~xor(1,1) → 1 & ~0 → 1 & 1 → 1

The order of precedence is important when using truth tables. Remember MATLAB's order of precedence when creating and evaluating logical statements. In the absence of parentheses, operators at the same level are read left to right.

DESCRIPTION	OPERATORS	
Parentheses	( )	Highest Priority
Transpose and Exponentiation	.' .^ ' ^	
Negation and Logical Negation	- ~	
Multiplication and Division	.* ./ .\ * / \	
Addition and Subtraction	+ -	
Relational Operators	< <= > >= == ~=	
Logical AND	&	
Logical OR	|	
Logical Short-Circuit AND	&&	Lowest Priority
Logical Short-Circuit OR	||	

Problem Steps

1. Open *PS08_truthtable_template.docx*. Complete the header, and save the file as **PS08_truthtables_*yourlogin*.docx**. This is where you will complete all the parts to this problem.

2. **To receive full credit, you must show the intermediate steps needed to reach the final answer for each location in each truth table—even if you see opportunities for short cuts**. See the example above.

   Note: you do not have to provide an answer for any greyed-out boxes within a table.

Problem Set 08
Relational and Logical Operators

3. On your Word answer sheet, do the following:

    a. Complete the truth table for the logic statement (B & ~A) & xor(A,B)

    b. Complete the truth table for the logic statement ~(L & M == ~N) | N & ~M > L

    c. Complete the truth table for the logic statement (~xor(F,G) & H) < K == G

    d. Construct the truth table and then complete it for the logic statement (E | C) & D | (D & E)

4. Parts (e) and (f) are English-language logic statements that need to be translated into MATLAB logic statements.

    e. The statement is true if both P and Q are true. The statement is true if both P and Q are false. The statement is true if R is true. All other conditions are false.

    f. The statement is true if one of S or T is true, or the statement is true if S is true and V is false.

    To translate English statements, follow these steps on your answer sheet:

    - Use the English language logic to determine the pattern of 1's and 0's in the truth table that must result from the statement.
    - Use the pattern to create the MATLAB logic statement.
    - Verify the MATLAB logic statement by completing the truth table.

Problem Set 08
Relational and Logical Operators

Problem 2: Indiana Corn Yield

Paired Programming

Problem Setup

Agricultural engineers can work in the area of crop production. The Indiana Field Office of the United States Department of Agriculture records crop yields for Indiana. When summarizing data, they divide Indiana into 9 regions, as seen in Figure 1. The Northern region comprises the northeast, north central, and northwest districts. The Central region comprises the east central, central, and west central districts. The Southern region comprises the southeast, south central, and southwest districts.

You have been provided with a data set, named **Data_corn_yield.txt**, of annual corn yields, in bushels per acre, for the nine districts that make up Indiana.

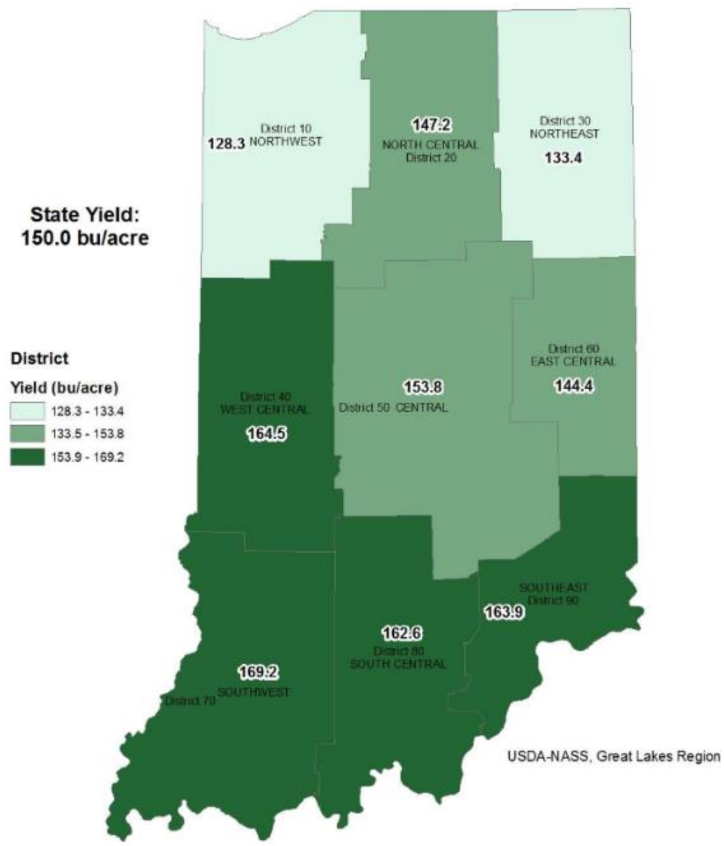

Figure 1: Indiana Agricultural Districts, with 2015 corn yields

Problem Steps

1. Use *PS08_yieldcompare_template.m* to write a no-input, no-output MATLAB function named **PS08_yieldcompare_*yourlogin1_yourlogin2*.m** that uses relational and logical operators and associated built-in functions, as appropriate, to answer the questions in Step 2.
2. Use the data to answer the following questions:
   A. How many years did the west central district have yields higher than the east central district, the central district, or the average yield for the whole Central region?
   B. In what year(s) did the west central district have higher yields than both the northwest district and the southwest district?
   C. In what year(s) did the Southern region have an average yield lower than the Central region average or the Northern region average? Hint: the problem setup defines the component districts for each region.
   D. How many years was the Northern region's average higher than the Southern region's average but lower than the Central region's average.
3. Display all answers clearly in the MATLAB Command Window. Do not hardcode any values in the `fprintf` statements.

   Hint: you can print vectors of values using `fprintf` or `disp`. Try these examples in MATLAB:

Problem Set 08
Relational and Logical Operators

```
vec = [1 2 3];
fprintf('The vector is:\n')
fprintf('%i\n',vec)
```

```
vec = [1 2 3];
fprintf('The vector is:\n')
disp(vec)
```

4. Publish your code to a PDF named **PS08_yieldcompare_*yourlogin1*_*yourlogin2*.pdf**.

Reference: https://www.nass.usda.gov/Statistics_by_State/Indiana/Publications/County_Estimates/index.php

Problem Set 08
Relational and Logical Operators

Problem 3: Thermocouples

Individual Programming

Problem Setup

You work for a company that manufactures thermocouples, which are small devices used to measure temperature (Figure 2). Thermocouples are designed to respond to changes in temperature, and their ability to respond quickly depends upon a thermocouple's materials, its geometry, and manufacturing processes.

Figure 2: Sample thermocouples.

A typical thermocouple response (in the time domain) is shown in Figure 3. From this time domain data, you can identify the 'time constant' which is a measure of how fast the thermocouple responds, measured in seconds. The time constant is the time it takes the response to reach 63.2% the maximum temperature difference. This is shown graphically on Figure 3, where the maximum value was 55 ^oC and the starting temperature was 50 ^oC.

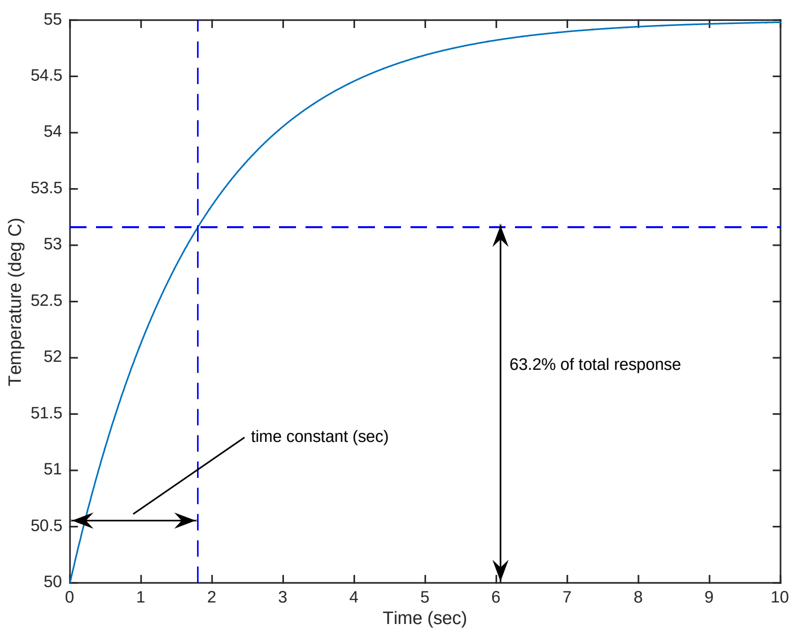

Figure 3: Thermocouple response and time constant identification

Your company has just completed manufacturing a new batch of 1,000 thermocouples, and a sub-set of those (*n* = 50) has been tested to determine their time constants. A determination must now be made to accept or reject those thermocouples based upon the criteria defined below.

Problem Set 08
Relational and Logical Operators

You have been provided with 5 time histories that define the boundaries of acceptability; see Figure 4. The experiment used to generate these times histories involved two large water tanks held at constant temperatures of 50 ^oC and 55 ^oC. The experiment involved moving the thermocouple quickly from the lower temperature tank to the higher temperature tank and capturing data from the thermocouple at very short time intervals (0.01 seconds).

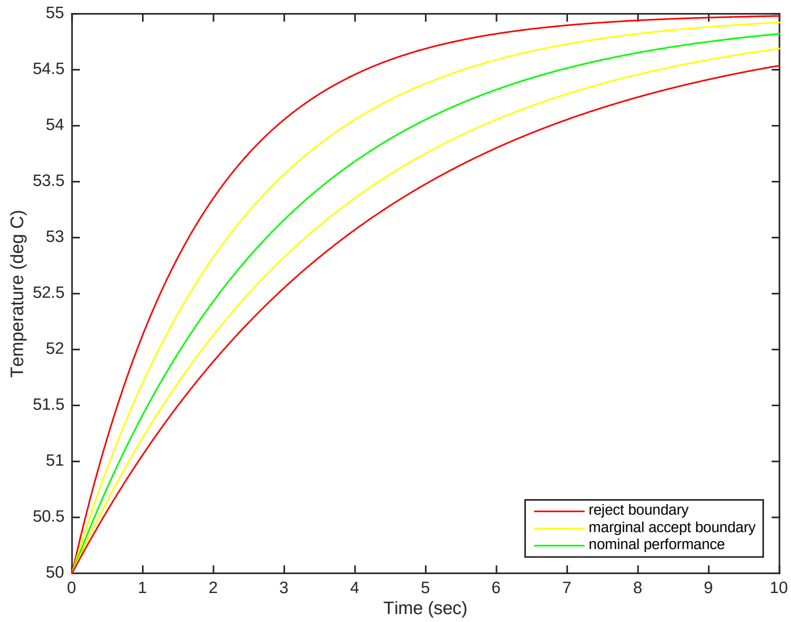

Figure 4: Performance boundaries for thermocouples

The 'nominal' performance is given in green, and this is the desired performance of each thermocouple. The two yellow time histories define the 'acceptability' boundary; thermocouple performance between those two yellow lines is acceptable. Performance between the yellow and red lines is 'marginally' acceptable, and performance outside the red lines is 'unacceptable' (those thermocouples will be rejected and will not be sold to customers).

The aim is to figure out whether a batch of product meets desired specifications. The 'reject' thermocouples are not necessarily bad because they have a fast or slow time constants, they are rejected because they do not meet the manufacturing specifications. Your task is to use the boundary data provided to establish boundaries between the 'reject' zone, the 'marginally accept' zone, and the 'accept' zone on the figure. Then, using those boundaries, decide whether to accept or reject each of the *n* = 50 thermocouples for which you have time constant data.

Two data files are available:

- **thermocouple_boundaries.txt** contains the time vector (first column) and 5 time histories (columns 2 - 6) corresponding to the boundaries of acceptability described above and in Figure 4.
- **time_constants.txt** contains the time constants for the *n* = 50 thermocouples.

Problem Set 08
Relational and Logical Operators

Problem Steps

1. Use PS08_thermocouple_template.m to create a no-input, no-output MATLAB user-defined function. Name your UDF file **PS08_thermocouple_*yourlogin*.m**. This UDF will
   a. Load the data files
   b. Generate a plot like Figure 4, with the 5 time histories (appropriately colored)
      i. On this plot, overlay the *n* = 50 time constants plotted using blue circles and no lines, where the x-coordinate is the time constant, and the y-coordinate is 63.2% of the maximum temperature difference.
   c. Use MATLAB to determine and report the number of thermocouples in each category (accepted, marginally accepted, rejected) to the MATLAB Command Window.
      i. Hint: compare your computed answer with the visual representation made in part b to understand if your results are making sense.
2. Publish your code to a PDF document named **PS08_thermocouple_*yourlogin*.pdf**.

Reference

Image source: http://www.meas-spec.com/product/t_product.aspx?id=7887