Problem Set 09 · For Loops

Instructions

1. Use the answer sheet provided in the Assignment Files to complete this problem set. Fill out the header information. Follow any additional instructions that appear in the answer sheet. Submit your finished answer sheet with the rest of your deliverables.
2. You will need your PS07_academic_integrity function for this assignment. You should make any fixes necessary to make it work properly.
3. Read each problem carefully before starting your work. You are responsible for following all instructions within each problem. Remember that all code submissions must follow the course programming standards.
4. Below are the expected deliverables for each problem.
   - Name your files to match the format in the table below.
   - Publish your code for each problem. See PS06 for more information.
   - Do not forget to include any data files loaded into your code.

Item	Type	Deliverable to include in Submission
Problem 1: Airspeed	Paired	☐ Flowchart (submitted in Answer Sheet)
	Individual	☐ Tracking Table (submitted in Answer Sheet) ☐ PS09_airspeed_*yourlogin*.m ☐ PS09_airspeed_*yourlogin*_report.pdf ☐ USAtmos_1976.p
Problem 2: Infinite Fin Revisit	Paired	☐ PS09_fin_revist_*login1*_*login2*.m ☐ PS09_fin_revist_*login1*_*login2*_report.pdf ☐ Flowchart (submitted in Answer Sheet)
Problem 3: Array Sorting	Individual	☐ PS09_sort_*login*.m ☐ PS09_sort _*login*_report.pdf ☐ Tracking Table (submitted in Answer Sheet)
PS09 Answer Sheet	Individual	☐ PS09_AnswerSheet_*yourlogin*.docx
Academic Integrity Statement	Individual	☐ PS07_academic_integrity_*yourlogin*.m

5. Save all files in a folder specific to PS09.
6. When you are ready to submit your assignment,
   - Compress all the deliverables into one zip file and name it **PS09_yourlogin.zip**. Be sure that you
     i. Only compress files using **.zip** format. No other compression format will be accepted.
     ii. Only include deliverables. Do **not** include the problem document, blank templates, etc.
   - Submit the zip file to the Blackboard drop box for PS09 before the due date.
7. After grades are released for this assignment, access your feedback via the assignment rubric in the My Grades section of Blackboard.

Problem 1: Airspeed

Individual

Learning Objectives

Your work on this problem may be assessed using any of the following learning objectives:

- Perform and evaluate algebraic and trigonometric operations
- Assign and manage variables
- Manage text output
- Perform and evaluate relational and logical operations
- Create and execute a user-defined function
- Track a flowchart with an definite looping structure
- Create test cases to evaluate a flowchart
- Create and troubleshoot a selection structure
- Convert between these definite looping structure representations: English, a flowchart, and code
- Code a definite looping structure that employs vector indexing
- Track execution of a definite looping structure that employs vector indexing using a variable tracking table

Problem Setup

Indicated airspeed (IAS) is a vital aircraft measurement. IAS is measured directly from pressure measurements taken from the aircraft exterior. One method to obtain these measurements is to use a pitot-static system.

A pitot tube is a hollow tube, bent at 90 degrees, that extends off the fuselage or wing of an aircraft. The center opening of the pitot tube points into the direction of the oncoming airflow and measures the total pressure. Another opening that is perpendicular to the airflow, either on the side of the pitot tube or on the aircraft fuselage, measures the static pressure. The difference between these two pressures is called the dynamic pressure. The dynamic pressure is directly related to the aircraft indicated airspeed through a version of Bernoulli's equation for compressible flow:

$$q = P\left[\left(1 + \frac{\gamma - 1}{2\gamma}\frac{\rho}{P}(Ma)^2\right)^{\frac{\gamma}{\gamma-1}} - 1\right]$$

Where

q is the dynamic pressure (kPa);

P is the static pressure at flight altitude (kPa) - in this problem, it is equal to the atmospheric pressure;

γ is a ratio between specific heats of air and is 1.4 for this application (dimensionless);

M is the Mach number (dimensionless);

a is the speed of sound at the flight level (m/s), which is calculated using

$$a = \sqrt{\gamma RT}$$

Where

R is the specific gas constant for air, 287.04 (N·m/(kg·K));

T = absolute temperature at flight altitude (K); and

ρ is the atmospheric density (kg/m^3), calculated using $\rho = P/RT$

To predict the atmospheric pressure and temperature at various altitudes, you will need to use an atmospheric model. The US Standard Atmosphere 1976 is a well-known model and you have access to a MATLAB version of the model, in a file named **USAtmos_1976.p**. This is a p-code file, which is encrypted and cannot be altered. It will calculate idealized atmospheric conditions at any altitude from sea level (0 km) and up to, but not including, 86 km using the US Standard Atmosphere 1976 model. The help lines for the code are supplied below.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Program Description
%   This function calculates the temperature and pressure of the
%   Earth's atmosphere at a given altitude, from 0 to below 86 km,
%   using the US Standard Atmosphere 1976 as the model.
%
% Function Call
%      [atm_pressure, atm_temperature] = USAtmos_1976(altitude)
%
% Input Arguments
% 1. altitude = altitude above sea level, 0 <= h < 86 (km) [scalar]
%
% Output Arguments
% 1. atm_pressure = atmospheric pressure (kPa) [scalar]
% 2. atm_temperature = atmospheric temperature (K) [scalar]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

You are an aerospace engineer designing an airspeed indicator that uses a pitot-static system. You plan to test your system, and you want to use Bernoulli's equation to predict the dynamic pressure at various atmospheric conditions before running any experiments. Your plan is to mimic the altitudes and velocity of crusing jet liners. Your first experiement will examine the pitot-static performance from altitudes of 20000 – 45000 ft at a constant Mach number of 0.85.

Your task is to write a user-defined function that will display information about the standard atmosphere and about the dynamic pressure and speed of sound at the experiment altitudes. Your function will accept no inputs and will return two outputs, one vector of of dynamic pressures and one vector of speeds of sound.

Your first step is to better understand the US Standard Atmosphere by visualizing the USAtmos_1976.p outputs over its full range of altitudes. You need to create one figure with two subplots where one subplot shows the atmospheric temperature every 0.5 kilometers from 0 to 85.5 km and the other shows the atmospheric pressure over the same altitude range. Note that it is convention to plot altitude on the y-axis when displaying atmospheric model information.

After you complete that, you will then use the standard atmosphere model and Bernoulli's equation to predict the dynamic pressure and speed of sound using an altitude vector of 20,000 ft, 28,000 ft, 32,000 ft, 36,000 ft, and 45,000 ft. You will store the dynamic pressure and speed of sound as vectors, which will be returned by the function as output arguments. You will also create two figures, one plotting dynamic pressure for each altitude and one plotting the speed of sound.

Problem Steps

1. **Before you start to code**: Create a flowchart to outline how information should move through the code.
    a. Open the answer sheet and examine the high-level flowchart.
    b. Create the two flowchart sections, one for the atmospheric pressure loop and one for the dynamic pressure and speed of sound loop.

  c. You can draw the flowchart using any means that result in a clear image for the answer sheet. Make sure your flowchart is legible.

  d. Add your flowchart sections to the answer sheet.

2. In your Answer Sheet, complete the tracking table for the dynamic pressure and speed of sound loop.

3. Use the high-level flowchart along with your flowchart sections to translate everything into a user-defined function. Use programming standards to place code in the appropriate sections within the template.

4. Test your function using the vector of altitudes provided for the tracking table. Run the function without suppressing the function call so that you can see the dynamic pressure and speed of sound results from the code.

5. Paste the function call and results displayed in the Command Window as comments under the `COMMAND WINDOW OUTPUTS` section of your function file.

6. Call your academic integrity function in the `ACADEMIC INTEGRITY` section.

7. Change the experimental altitude vector to be every 1000 ft from 20,000 to 45,000 ft.

8. Publish your function to a PDF and name the published file as required in the deliverables list. Suppress your function call when you publish the function.

References

https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19800015804.pdf

https://www.grc.nasa.gov/www/k-12/airplane/pitot.html

Problem 2: Infinite Fin Revisit

Paired

Learning Objectives

Your work on this problem may be assessed using any of the following learning objectives:

- Perform and evaluate algebraic and trigonometric operations
- Assign and manage variables
- Manipulate arrays (vectors or matrices)
- Manage text output
- Perform and evaluate relational and logical operations
- Create and execute a user-defined function
- Construct a flowchart for nested structures using standard symbols and pseudocode
- Construct a flowchart using standard symbols and pseudocode
- Create and troubleshoot a selection structure
- Code a definite looping structure that employs vector indexing
- Code an indefinite looping structure
- Code nested looping structures that employ array indexing

Problem Setup

In PS08 Problem 2, you wrote code to determine the minimum length of a rod used to cool a heat source. For this problem, you change the function you wrote to accept different input arguments that will allow you to vary the rod diameter to determine the minimum length of a rod for a set of rod diameters. Your new function must keep all the PS08 Problem 2 requirements but with the following changes:

- Accept 3 input arguments: minimum diameter value, maximum diameter value, and thermal conductivity value
- Add the heat source temperature and ambient air temperature as assigned variables within the code
- Increment the rod diameter by 0.5 mm
- Display the rod diameter along with its minimum length to the Command Window
- Return 1 output argument: a vector of minimum rod lengths for each rod diameter

Problem Steps

1. Reread PS08 Problem 2. Both you and your paired partner should have your own version of this code. Examine your solutions, select one file to use for this problem (you will need both the code and the flowchart), and make any necessary corrections to the calculations and print commands.
2. Adapt your PS08 flowchart to make it include the new functionality required by this problem as well as any changes need from step 1.
3. Test your function by calling it with a minimum rod diameter of 1 mm to 10 mm for aluminium.

   Do not suppress your function call in the Command Window. This will allow the output argument display to the Command Window along with the printed display. Paste the function call and results displayed in the Command Window as comments under the `COMMAND WINDOW OUTPUTS` section of your function file.
4. Call your academic integrity function in the `ACADEMIC INTEGRITY` section.
5. Publish your function as a PDF file using <u>any valid test case</u> and name the file as directed in the deliverables list. Suppress your function call while publishing.

Problem 3: Array Sorting

Individual

Learning Objectives

Your work on this problem may be assessed using any of the following learning objectives:

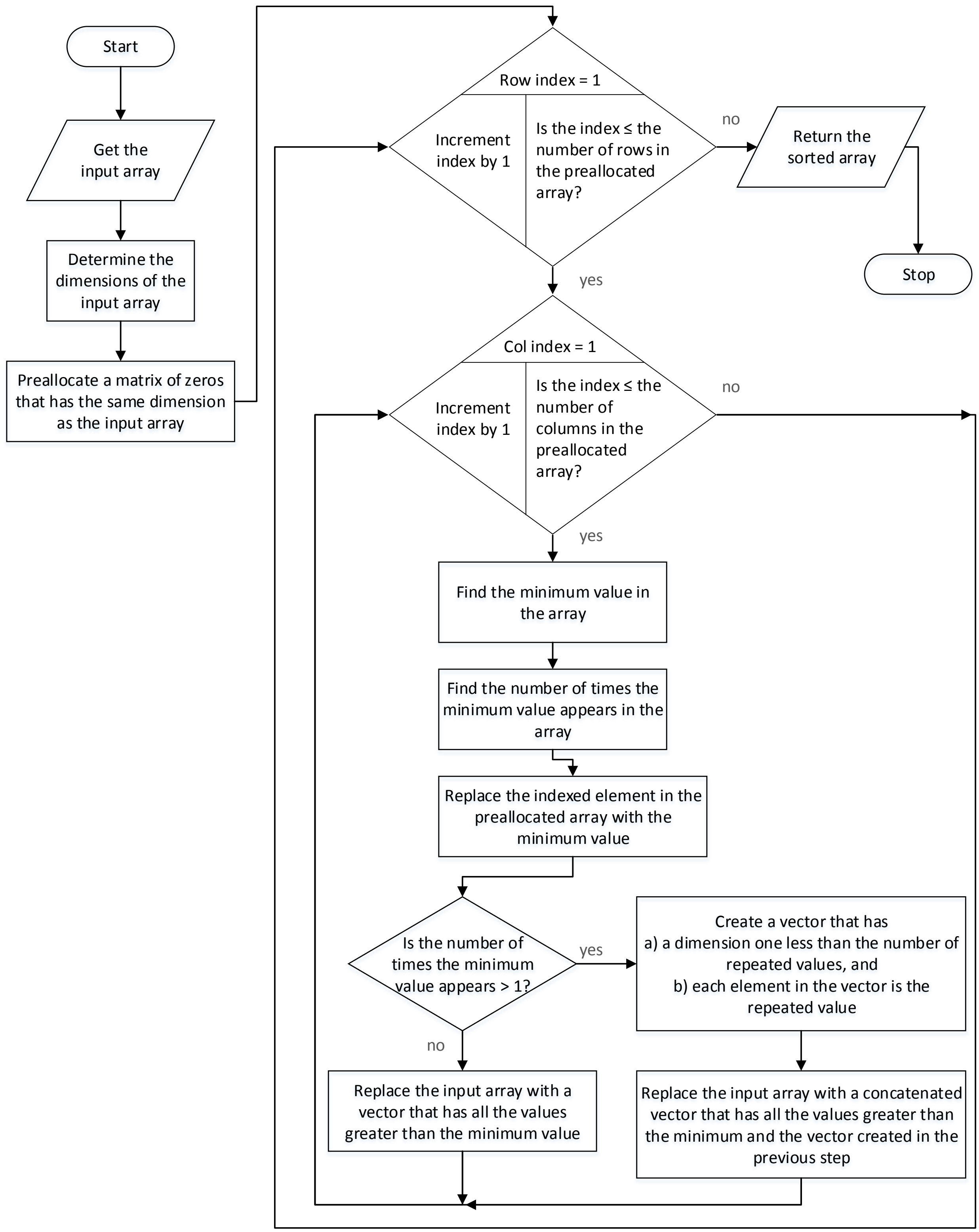
- Perform and evaluate algebraic and trigonometric operations
- Assign and manage variables
- Manage text output
- Perform and evaluate relational and logical operations
- Create and execute a user-defined function
- Track a flowchart with an definite looping structure
- Create test cases to evaluate a flowchart
- Create and troubleshoot a selection structure
- Create and troubleshoot a repetition structure
- Convert between these nested structures representations: English, a flowchart, and code
- Code nested looping structures that employ array indexing
- Track nested structures using a variable tracking table

Problem Setup

There are many sorting algorithms in computer programming. MATLAB has some built-in sorting capabilities, but those commands may not always fit your needs. For this problem, you must rearrange the elements in an array so that the smallest value moves into is in the upper left corner of the array, the next smallest value moves to the first row second column, and so on until the largest value is in the lower right corner of the array, as shown in the example below:

$$A = \begin{bmatrix} 5 & 2 & -1 \\ 2 & 6 & 3 \\ 9 & -2 & 0 \end{bmatrix} \rightarrow A_{sorted} = \begin{bmatrix} -2 & -1 & 0 \\ 2 & 2 & 3 \\ 5 & 6 & 9 \end{bmatrix}$$

You must use the following flowchart to write a function that will accept an array of any size and sort it in the desired manner. Translate this specific flowchart into MATLAB code.


Start
Get the input array
Determine the dimensions of the input array
Preallocate a matrix of zeros that has the same dimension as the input array
Row index = 1
Increment index by 1
Is the index ≤ the number of rows in the preallocated array?
no
Return the sorted array
Stop
yes
Col index = 1
Increment index by 1
Is the index ≤ the number of columns in the preallocated array?
no
yes
Find the minimum value in the array
Find the number of times the minimum value appears in the array
Replace the indexed element in the preallocated array with the minimum value
Is the number of times the minimum value appears > 1?
yes
Create a vector that has
a) a dimension one less than the number of repeated values, and
b) each element in the vector is the repeated value
no
Replace the input array with a vector that has all the values greater than the minimum value
Replace the input array with a concatenated vector that has all the values greater than the minimum and the vector created in the previous step

Problem Steps

1. **Before you start to code**: Review the flowchart to understand the process for sorting the array.
2. In your Word answer sheet, complete the variable tracking table by hand for the execution of the nested loops using the provided input argument.
3. Translate the flowchart above to a MATLAB user-defined function using the proper template. Comment your code appropriately and follow the ENGR132 Programming Standards.
4. Test your function by calling it with the test case provided in Step 2. Add the following test cases:
   a. $B = \begin{bmatrix} 100 & -72 & 14 & 30 & 27 \end{bmatrix}$
   b. $C = \begin{bmatrix} 2 & 0.5 & -5 & 3 & 6 \\ -5 & 4 & -3 & 4 & 6 \\ 8 & 2.5 & 1 & -2 & -1 \end{bmatrix}$

   Do not suppress the output when you call your function. Paste the function call and results displayed in the Command Window as comments under the `COMMAND WINDOW OUTPUTS` section of your function file.
5. Call your academic integrity function in the `ACADEMIC INTEGRITY` section.
6. Publish your function as a PDF file using any valid test case and name the file as directed in the deliverables list. Suppress your function call while publishing.