Problem Set 07 · Selection Structures

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. 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.
3. 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 Assignment Files for help with publishing.
    - Do not forget to include any data files loaded into your code.

Item	Type	Deliverable to include in Submission
Problem 1: Security Camera Placement	Paired	☐ PS07_observatory_*login1*_*login2*.m ☐ PS07_observatory_*login1*_*login2*_report.pdf
Problem 2: Academic Integrity Statement	Individual	☐ PS07_integrity_*yourlogin*.m ☐ PS07_integrity_*yourlogin*_report.pdf
Problem 3: Storage Tank Volume	Individual	☐ PS07_tankVolume_*yourlogin*.m ☐ PS07_tankVolume_*yourlogin*_report.pdf
PS07 Answer Sheet	Individual	☐ PS07_answer_sheet_*yourlogin*.docx

4. Save all files to a folder specific to PS07.
5. When you are ready to submit your assignment,
    - Compress all the deliverables into one zip file and name it **PS07_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 PS07 before the due date.
6. After grades are released for this assignment, access your feedback via the assignment rubric in the My Grades section of Blackboard.

Notes Before You Begin this Assignment

Helpful MATLAB Commands

Learn about the following built-in MATLAB commands, which might be useful in your solutions:

`if, elseif, else, end, isstring, acos`

Problem 1: Observatory Security Cameras

Paired

Learning Objectives

Below are learning objectives that may be used to assess your work on this problem. Learning objectives from past assignments may also be used to assess your work. Use the links to find the full evidence lists for each topic.

Variables	02.00 Assign and manage variables
Text Display	05.00 Manage text output
Relational & Logical Operators	14.00 Perform and evaluate relational and logical operations
User-Defined Functions	11.00 Create and execute a user-defined function
Flowcharts	15.01 Construct a flowchart for a selection structure using standard symbols and pseudocode
	15.02 Track a flowchart with a selection structure
	15.09 Create test cases to evaluate a flowchart
	15.10 Construct a flowchart using standard symbols and pseudocode
Selection Structures	16.01 Convert between these selection structure representations: English, a flowchart, and code
	16.02 Code a selection structure

Problem Setup

An astronomical society restored their historic observatory and converted it to a small museum. They hired your engineering firm to design a security system for the updated building. The building's layout, with pertinent measurements but not drawn to scale, is shown in the figure.

The society wants the ability to place security cameras wherever they choose in each of the rooms. At night, all the doors will be closed and the cameras will be turned on.

Each camera will communicate its coordinates to a central receiver. Code must be written to confirm the location of the camera based on its reported coordinates. The building's coordinate system has its origin in the center of the observatory. Cameras must be on the ceiling inside a room; they cannot be on any wall or door or outside the building.

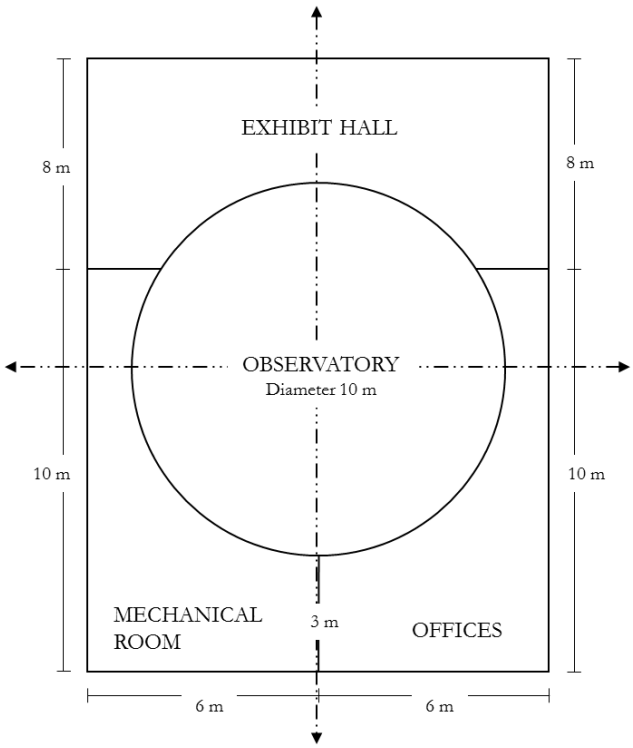


Figure 1. Museum floor plan

You must create a function to ensure that each camera has valid coordinates, regardless of its location within a room. Your function must accept the x and y coordinates of a camera's location. It must display the x and y coordinates along with the location of the camera or that the location is invalid. It should not return any output arguments.

Problem Steps

Pairs WY and XZ work together on a flowchart.

1. Create a flowchart to outline how information should move through the code.
    a. Draw a flowchart that:
        i. Will state the correct location of a pair of x, y coordinates that are inside the
            1. observatory,
            2. exhibit hall,
            3. mechanical room, or
            4. offices
        ii. Will state that a pair of x, y coordinates that are on a wall or outside the building are invalid coordinates
        iii. Uses proper flowchart symbols and guidelines for selection structures. See LO 15.10 and LO 15.01 for help.
    b. You can draw the flowchart using any means that result in a clear image for the answer sheet. Make sure your flowchart is legible. Options include
        - Drawing it by hand and taking a clear photo
        - Drawing it directly in the Word answer sheet using Word's drawing tools
        - Drawing it in Microsoft's Powerpoint, Publisher, or Visio
        - Using another flowchart tool, such as Lucidchart
2. Open **ONE** copy of PS07_AnswerSheet_login.docx. In the Answer Sheet,
    a. Paste a clear image of your flowchart
    b. Complete the Test Cases table using your flowchart.

Pairs WZ and XY work together on the code using only ONE flowchart.

3. Establish which flowchart to use as the basis for the code:
    a. For pair WZ: you both must use W's flowchart and answer sheet.
    b. For pair XY: you both must use X's flowchart and answer sheet.
4. Translate the flowchart to a MATLAB user-defined function. Name your UDF file as required in the Deliverables List. Your function must
    a. accept two input arguments: the x and y coordinates of the camera location
    b. return no output arguments
    c. display the location of the camera coordinates or an error message to the Command Window
5. Test your function using the test cases that you specified in Step 2.
6. For each test case, paste the **function call** and **results displayed** in the Command Window as comments under the `COMMAND WINDOW OUTPUTS` section of your function file.

7. Now that your code is functional, do you think that the flowchart and test cases you established in Step 2 were sufficient to completely test all possible paths in the structure that this problem requires?
    a. If not, run additional test cases through your code to thoroughly test all possible paths in your code and paste the function call and results into the `COMMAND WINDOW OUTPUTS` section of your function file, as you did in step 6.
    b. Examine how well your code follows your flowchart. On the Answer Sheet's Flowchart and Code Comparison section, explain any differences between the code and the flowchart, and state why those changes were necessary.
8. Save a copy of the answer sheet for both students in the pair. You will each need it to complete Problems 2 and 3 individually.
9. Publish your function to a PDF using any valid input arguments and name the published file as required in the deliverables list.

Problem 2: Academic Integrity Statement

Individual

Learning Objectives

Below are learning objectives that may be used to assess your work on this problem. Learning objectives from past assignments may also be used to assess your work. Use the links to find the full evidence lists for each topic.

Variables	02.00 Assign and manage variables
Text Display	05.00 Manage text output
Relational & Logical Operators	14.00 Perform and evaluate relational and logical operations
User-Defined Functions	11.00 Create and execute a user-defined function
Flowcharts	15.02 Track a flowchart with a selection structure
	15.09 Create test cases to evaluate a flowchart
	15.10 Construct a flowchart using standard symbols and pseudocode
Selection Structures	16.01 Convert between these selection structure representations: English, a flowchart, and code
	16.02 Code a selection structure

Problem Setup

At the bottom of all ENGR 132 script and function templates is an Academic Integrity statement. In this problem, you will create a function that will allow you to display your and your team's names along with the academic integrity statement.

MATLAB allows you to create a string array, which is an array where each element in the array is a string instead of a number. It is important that you use proper formatting in a string array. Enter this command into MATLAB:

```
purdue_astronauts = ["Neil Armstrong","Eugene Cernan","Loral OHara","Scott Tingle"]
```

The double quotes are required; single quotes will not create a string array. You can use vector commands, such as **`length`** and **`size`**, on string arrays.

For this problem, you will create a function that will display the following academic integrity statement for an individual problem:

> I am submitting code that is my own original work. I have not used source code, either modified or unmodified, obtained from any unauthorized source. Neither have I provided access to my code to any peer or unauthorized source. Signed,
>
> *<your name>*

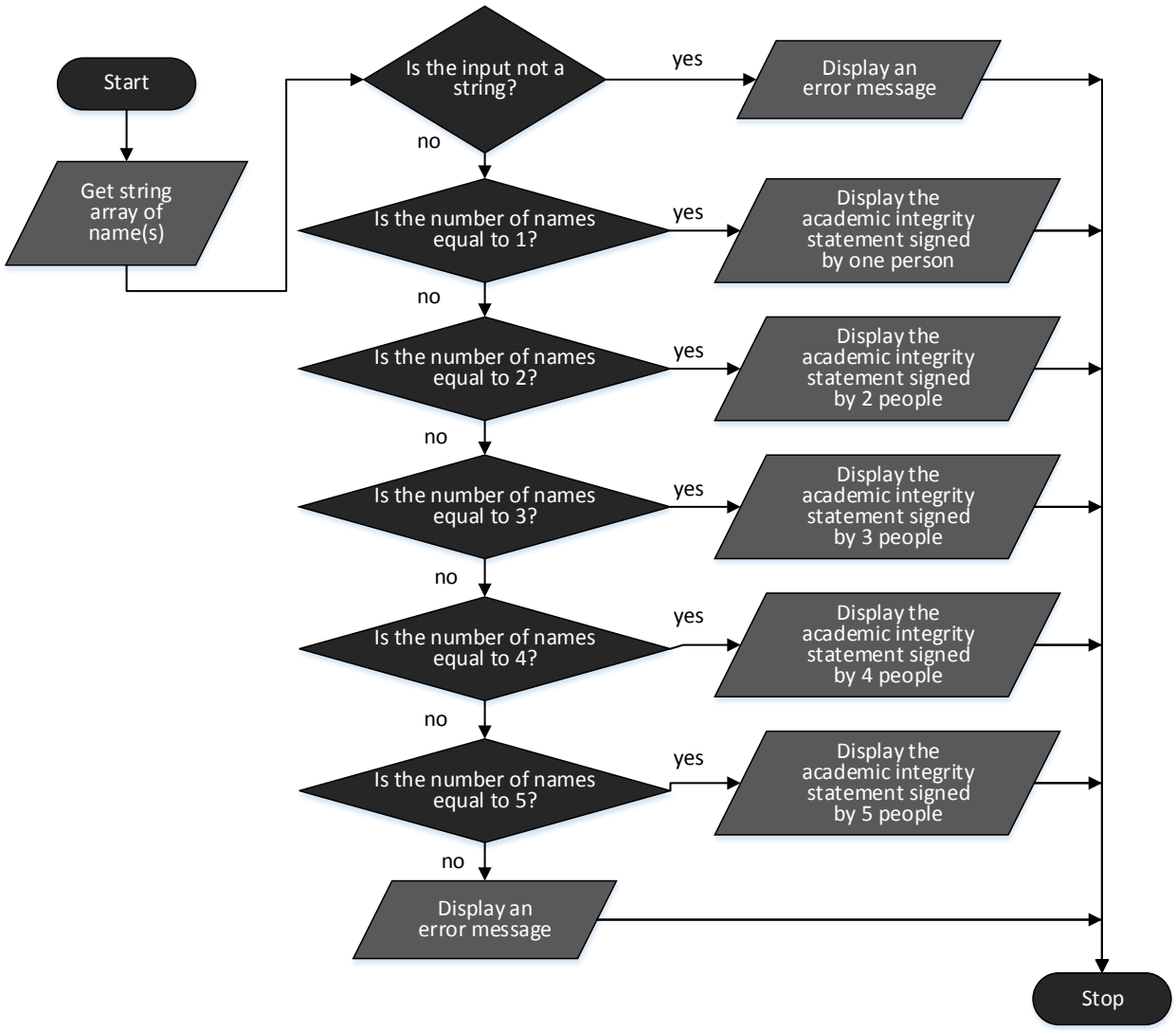
For a paired or team submission, your function will display a plural version of the academic integrity statement:

> We are submitting code that is our own original work. We have not used source code, either modified or unmodified, obtained from any unauthorized source. Neither have we provided access to our code to any peer or unauthorized source. Signed,
>
> *<Student 1>*

```
<Student 2>
<Student 3>
<Student 4>
```

where the number of student names displayed equals the number of student names in the input argument array. The function must be able to handle one, two, three, four, or five names. It must provide a meaningful error message for improper inputs. You will call this function in all assignments from this point forward.

Your function must follow this flowchart:



Problem Steps

1. **Before you start to code:** Review the flowchart to understand the process for which academic integrity statement gets displayed to the Command Window.
2. In your Answer Sheet:
    a. Add a series of test cases to thoroughly examine all the possible paths in the flowchart.

   b. Record the corresponding flowchart outputs. These outputs must be determined independently of MATLAB (using the flowchart) so that they provide a means to check and debug your MATLAB code as you write the code.

3. Translate the flowchart to a MATLAB user-defined function using the appropriate function template. Your function must
   a. accept one input argument: a string array of names;
   b. return no output arguments;
   c. display the academic integrity statement with only the relevant names for the assignment;
   d. display a warning message if the number of names doesn't match an expected value;
   e. display a warning message if the input argument is not a string.

   Hint: Use MATLAB to learn about the `isstring` command.

4. Run the function using each of the test cases on your answer sheet. For each test case, paste the **function call** and **results displayed** in the Command Window as comments under the `COMMAND WINDOW OUTPUTS` section of your function file.
5. Publish your function to a PDF using your name and your paired partner's name and name the published file as required in the deliverables list.

Problem 3: Storage Tank Volume

Individual

Learning Objectives

Below are learning objectives that may be used to assess your work on this problem. Learning objectives from past assignments may also be used to assess your work. Use the links to find the full evidence lists for each topic.

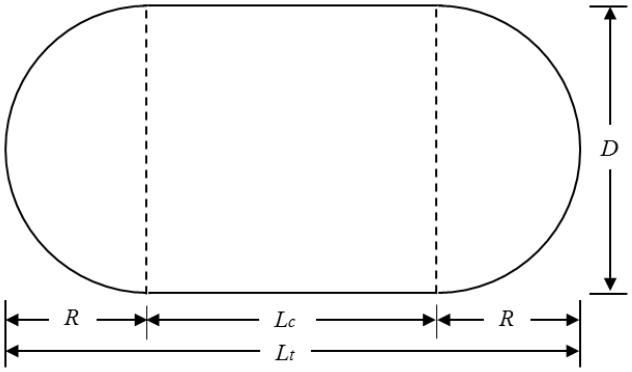
Calculations	01.00 Perform and evaluate algebraic and trigonometric operations
Variables	02.00 Assign and manage variables
Text Display	05.00 Manage text output
Relational & Logical Operators	14.00 Perform and evaluate relational and logical operations
User-Defined Functions	11.00 Create and execute a user-defined function
Flowcharts	15.01 Construct a flowchart for a selection structure using standard symbols and pseudocode
	15.02 Track a flowchart with a selection structure
	15.09 Create test cases to evaluate a flowchart
	15.10 Construct a flowchart using standard symbols and pseudocode
Selection Structures	16.01 Convert between these selection structure representations: English, a flowchart, and code
	16.02 Code a selection structure

Problem Setup

Cylindrical steel tanks have many uses. They are used in oil and gas refining, food production, farming, liquified gas storage, and more. Cylindrical tanks can have flat end caps, elliptical end caps, or spherical end caps.

Your company uses cylindrical tanks with spherical end caps. The tanks are sometimes installed horizontally and sometimes vertically. You are designing a fluid level measurement system for the tanks. A probe will measure the height of the fluid in the tank, and that fluid height will be used to determine the volume of the liquid in the tank. The system needs to work for both horizontal or vertical tanks.

The tanks have the same geometry, whether installed horizontally or vertically. The two hemispherical end caps are equivalent to one sphere. The end caps and the cylindrical center section have the same radius, R. The tank length, L_t, is the sum of the tank diameter, D, and the length of the cylindrical center, L_c. All lengths are interior measurements. The tank wall thickness is not required.



When a tank is installed horizontally, the fluid volume at any fluid height within the tank can be calculated using the equation

$$V_f = \frac{\pi h^2(3R - h)}{3} + L_c\left(R^2 \cos^{-1}\left(\frac{R-h}{R}\right) - (R-h)\sqrt{2Rh - h^2}\right)$$

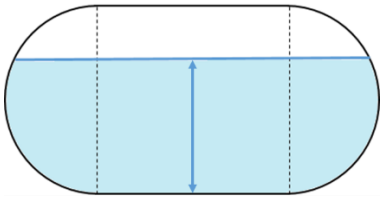
Where

$V_f =$ fluid volume

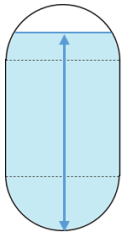
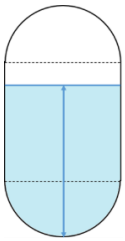
$h =$ fluid height (between 0 and the tank diameter, inclusive of both; measured from the tank bottom)

$R =$ tank radius

$L_c =$ length of cylindrical section of the tank



When a tank is installed vertically, the fluid volume at a given fluid height requires different equations. The proper equation to use will be determined by the height of the fluid in the tank, where the fluid height is measured from the bottom of the tank.

Criteria	**Equation**	**Example Image**
The tank's lower hemisphere and center cylinder are both full; the top hemisphere is at least partially full. Valid fluid height, h, for this equation: $(R + L_c) \leq h \leq (2R + L_c)$	$V_f = \pi R^2 L_c + \frac{\pi(h - L_c)^2}{3}(3R - h + L_c)$	
The tank's lower hemisphere is full and the center cylinder section is partially or completely full; the top hemisphere is empty. Valid fluid height, h, for this equation: $R \leq h \leq (R + L_c)$	$V_f = \frac{2\pi R^3}{3} + \pi R^2(h - R)$	
The tank's lower hemisphere is somewhere between empty and full. Valid fluid height, h, for this equation: $0 \leq h \leq R$	$V_f = \frac{\pi h^2}{3}(3R - h)$	

Your task is to write a user-defined function that will calculate the fluid volume for any height in a tank that has a diameter of 3.35 meters, a length of 21.1 meters, and either a vertical or horizontal orientation. Your UDF must

- Accept two (2) inputs: tank orientation and fluid height within the tank.
- Returns one (1) output: fluid volume
- If either the fluid height or the tank orientation is not valid, return a fluid volume of -1 and print a useful warning to the Command Window.

Problem Steps

1. **Before you start to code**: Create a flowchart to outline how information should move through the code.
    a. Draw a flowchart to:
        i. Check for valid fluid height and tank orientation,
        ii. Calculate the tank volume.
    b. You can draw the flowchart using any means that result in a clear image for the answer sheet. Make sure your flowchart is legible. Options include:
        - Drawing it by hand and taking a clear photo
        - Drawing it directly in the Word answer sheet using Word's drawing tools
        - Drawing it in Microsoft's Powerpoint, Publisher, or Visio
        - Using another flowchart tool, such as Lucidchart
2. In your Answer Sheet,
    a. Paste a clear image of your flowchart
    b. Select a series of test cases to thoroughly test all possible paths on your flowchart
    c. Record the fluid volume or the error for each test case
3. Translate your flowchart into a user-defined function using the appropriate template. Use programming standards to place code in the appropriate sections within the template.

    Hint: Coding large equations in one line is difficult. Calculate smaller terms and assign them to MATLAB variables; then build the equation from the assigned variables and any remaining terms.

    Example: Code $x = \frac{-b+\sqrt{b^2-4ac}}{2a}$ using assigned scalar variables a, b, and c

    ```
    % CALCULATIONS
    discrim = sqrt(b^2 - 4*a*c);
    denom = 2*a;
    x = (-b + discrim)/denom;
    ```
4. Test your function using the test cases that you specified in Step 2. Do not suppress your function call while running the test cases.
5. For each test case, paste the function call and resulting information in the Command Window as comments under the `COMMAND WINDOW OUTPUTS` section of your function file.
6. Now that your code is functional, do you think that your flowchart and the test cases you established in Step 2 were sufficient to completely test all possible paths in the structure that this problem requires?
    a. If not, run additional test cases through your code to thoroughly test all possible paths in your code and paste the function call and results into the `COMMAND WINDOW OUTPUTS` section of your function file, as you did in step 5.
    b. Examine how well your code follows your flowchart. On the Answer Sheet's Flowchart and Code Comparison section, explain any differences between the code and the flowchart, and state why those changes were necessary.
7. Call your academic integrity function from Problem 2 in the `ACADEMIC INTEGRITY` section
8. Publish your function to a PDF using any valid set of inputs and name the published file as required in the deliverables list.