A08 · Repetition Structures

Introduction

Assignment Goals

This assignment focuses on two repetition structures: while loops and for loops. You will gain programming experience with loops and then you will apply those skills to an engineering context. You will learn to differentiate loop types in flowcharts. You will also practice professional communication and plot displays.

Successful Completion

This assignment has **4** problems. The deliverables list contains everything you are expected to submit individually.

Submit Problems 1 and 2 to the Gradescope online assignment **A08 – Skills Problems**		
Problem	**Type**	**Deliverables**
Problem 1: MATLAB Skills – While Loops	Individual	□ A08Prob1_while_*login*.m □ Requested results and information
Problem 2: MATLAB Skills – For Loops	Individual	□ A08Prob2_for_*login*.m □ Requested results and information
Submit Problems 3 and 4 to the Gradescope programming assignment **A08 – Context Problems**		
Problem	**Type**	**Deliverables**
Problem 3: Tank Volume	Individual	□ A08Prob3_tankVol_*login*.m □ Supporting file(s): ○ A08Prob3_figure_*login*.png
Problem 4: Exercise Schedule	Individual	□ A08Prob4_exercise_*login*.m

1. Read *Notes Before You Start*, on **Page 2**.
2. Read each problem carefully. You are responsible for following all instructions within each problem.
    a. Problems 1 and 3 require while loops; problems 2 and 4 require for loops. They are in this order to make submission to Gradescope easier.
    b. Problems 3 and 4 each have a team planning component that you are expected to submit separately from your deliverables. Read the instructions within each of those problems.
3. Complete the problems using the problem-specific m-file templates provided in the assignment download. Replace *template* in the filename with your Purdue Career Account login
4. When your work is complete, confirm your deliverables and your team plans are submitted to Gradescope.

Learning Objectives & Grading

This course uses learning objectives (LOs) to assess your work. You can find a list of the course LOs on Brightspace (Content > Key Course Info > Learning Objectives). Review the assignment grading for each problem in this assignment, which starts on **Page 10**. This outlines how your work will be graded for each problem.

Notes Before You Start

Helpful MATLAB Commands

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

```
acos, while, for, end, input, disp, sqrt
```

Coding Long Expressions

Coding large expressions or commands in one line is difficult. You have two options to manage statement length.

Build from smaller terms

One way is to calculate smaller terms and assign them to MATLAB variables; then build the expression 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;
```

MATLAB ellipsis

You can also use MATLAB's ellipsis (`...`) to break up long lines of code for readability. Read this MATLAB documentation for more guidance on using the ellipsis: Continue Long Statements on Multiple Lines

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

```
% CALCULATIONS
x = (-b + sqrt(b^2 - 4*a*c))...
    /(2*a);
```

Gradescope

You will submit all your deliverables to Gradescope for grading. This homework has **two** Gradescope submission assignments for the individual submissions, plus a **third** assignment for the team planning component:

- **A08 – Skills Problems**: submit your deliverables for Problems 1 and 2
- **A08 – Context Problems**: submit your deliverables for Problems 3 and 4
- **A08 – Team Planning**: submit your team plans for Problems 3 and 4 as a team

Problem Generator File

In the assignment folder, you will see a file named **A08_skills.p**. This is a MATLAB function file that generates problem information for each skill problem in this assignment.

Need help using this file? Refer to the *Notes Before You Start* of A04 or A05, or review the A00 Activity from Class 1B to see fully-worked examples.

Problem 1: MATLAB Skills – While Loops

Introduction

This problem allows you to practice writing a while loop repetition structure. You will submit your answers to an online assignment on Gradescope.

Problem Instructions

1. Type this command into the MATLAB Command Window prompt:

   ```
   >> A08_skills(PUID, 1)
   ```

   Remember to replace PUID with your 8-digit Purdue University ID number (leave off the leading 00).
2. Read the written instructions that appear in the Command Window. Use MATLAB to check your answers.
   a. Use the included template to write the requested code.
   b. Run your script to test your loop and generate the requested information.
   c. Programming standards will not be assessed in this problem. Do not include comments in your solutions. Properly name the file.
3. Submit your work in Gradescope:
   a. Open Gradescope > **A08 – Skills Problems** and find the set of boxes that belong to **Q1** you want to submit. Enter the required information along with your answer:

      □ **Function call**. Copy the command that you entered at the command prompt to call the function and paste the full command into this box. Be sure your PUID is included.

      □ **Instruction text**. Copy the instruction text that is displayed in the Command Window. Paste it into this box. Include all text provided.

      □ **Solutions**. Enter your solutions and script. Follow any additional instructions provided.

   b. When you have entered all the required information for the question, click the **Save Answer** button.

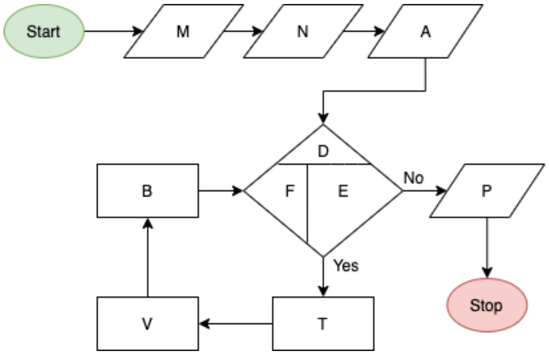
Problem 2: MATLAB Skills – For Loops

Introduction

This problem allows you to practice translating a `for` loop flowchart into code. You will also practice using a control vector in your `for` loop that does not have sequential values. You will submit your answers to an online assignment on Gradescope.

Problem

Below is a blank flowchart that only contains letters in the shapes.



To get the instructions for this flowchart, you will need to use your p-code file, **A08_skills.p**. The instruction text will provide you with a control vector, Z, to use in the `for` loop and will provide the instruction text for each shape in the flowchart. Match the text with the corresponding letters in the flowchart.

Instructions

1. Type this command into the MATLAB Command Window prompt:

```
>> A08_skills(PUID, 2)
```

   Remember to replace PUID with your 8-digit Purdue University ID number (leave off the leading 00).

2. Read the instructions and conditions that display to the Command Window.
   - Use the included template to write the requested code.
   - Maintain the order of the items in the flowchart. Even if there is a different approach you could take to the solve the problem, you must follow the exact order and instructions in the flowchart.
   - You should solve this problem **without** using vector indexing in the loop calculations.
   - Programming standards will not be assessed in this problem. Use the single-letter variables used in the instruction text. Do not include comments in your solutions. Properly name your script.

3. Submit your work in Gradescope:
   a. Open Gradescope > **A08 Skills** and find the set of boxes that belong to **Q2** you want to submit. Enter the required information along with your answer:
      - ☐ **Function call**. Copy the command that you entered at the command prompt to call the function and paste the full command into this box. Be sure your PUID is included.
      - ☐ **Instruction text**. Copy the instruction text that is displayed in the Command Window. Paste it into this box. Include all text provided.
      - ☐ **Solutions**. Enter your solutions and script. Follow any additional instructions.

   b. When you have entered all the required information for the question, click the **Save Answer** button.

Problem 3: Tank Volume

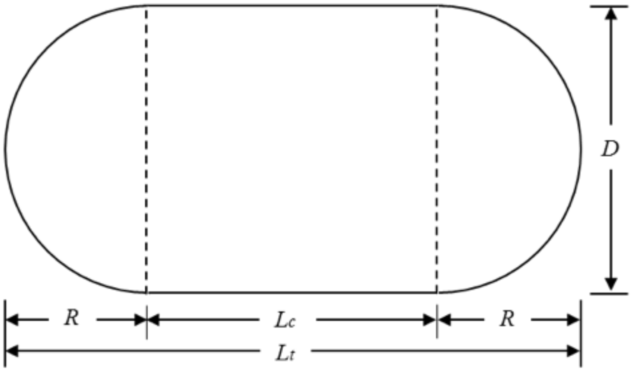
Introduction

This problem focuses on your ability to translate a flowchart into a MATLAB script that uses a while loop to answer engineering-based questions. Be sure to follow good programming standards in your script.

Problem

Cylindrical steel tanks have many outdoor 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. You are working on a design for a tank fill measurement system that can be used in tanks that are installed horizontally. 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 tanks have the geometry shown in the figure. 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 for this application.

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

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

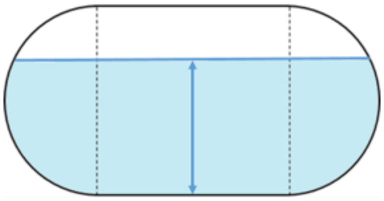
Where

$V_f =$ 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



It is important to never overfill a tank. Liquid in outdoor industrial tanks can expand with ambient air temperature changes, so the tank must be empty enough to accommodate the expansion. Your measurement system must ensure that the fill process will shut down properly and not allow a tank to overfill. The first step to achieve this is to set a safety percent, which is the maximum percentage of the tank volume that can be filled.

The fluid height is not measured instantaneously in this system. Instead, the programmer sets an increment and the fluid height is measured in those increments. For example, if the increment is 0.5m and the tank is empty at the start of the fill, then the volume will be calculated at 0m, 0.5m, 1.0m, 1.5m, etc. until the tank is properly filled.

To ensure the tank does not overfill, you must stop the fill before the fluid volume passes the safety volume. You will keep the volume from exceeding the safety limit by setting a maximum tolerance. To find the maximum allowable tolerance, V_{tol}, you will calculate the difference in fluid volume at two different fluid heights, $R + 0.5\Delta h$ and $R - 0.5\Delta h$:

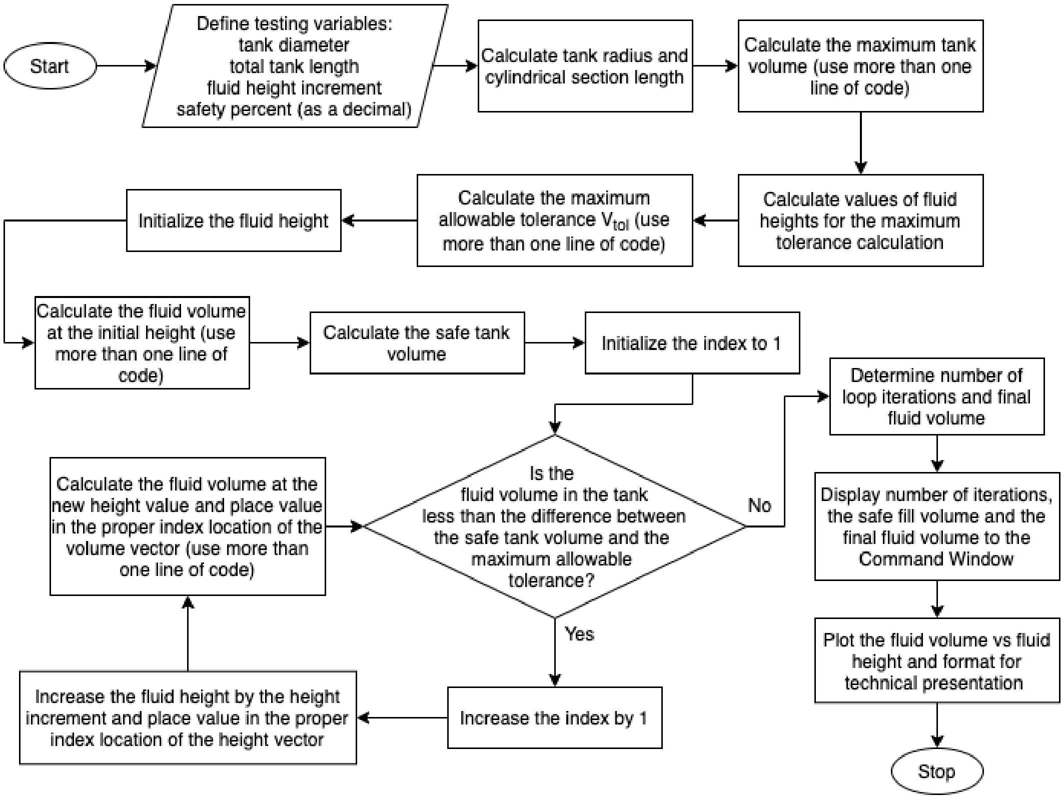
$$V_{tol} = V_f(R + 0.5\Delta h) - V_f(R - 0.5\Delta h)$$

Using the volume function V_f above, where Δh is the fluid height increment and R is the tank radius.

For testing purposes, you will use your company's most popular tank size, which has a diameter of 4.1 meters and a total tank length of 20.5 meters. Assume a fluid height increment of 0.25 meters and that the tank can be safely filled to 80% capacity (0.8 as a decimal). Also assume that the tank is completely empty when filling begins.

To meet the safety needs of the system, you will write a script that translates the flowchart below. Your script will

- Define the testing values as variables,
- Calculate the requested values,
- Create a vector of fluid height values and another vector of corresponding fluid volumes using a while loop,
- Stop adding elements to the vector as close as possible to the fill capacity without exceeding that value,
- Plot the resulting vectors as fluid volume versus fluid height. Format for technical presentation.


Start
Define testing variables: tank diameter total tank length fluid height increment safety percent (as a decimal)
Calculate tank radius and cylindrical section length
Calculate the maximum tank volume (use more than one line of code)
Calculate values of fluid heights for the maximum tolerance calculation
Calculate the maximum allowable tolerance V_{tol} (use more than one line of code)
Initialize the fluid height
Calculate the fluid volume at the initial height (use more than one line of code)
Calculate the safe tank volume
Initialize the index to 1
Is the fluid volume in the tank less than the difference between the safe tank volume and the maximum allowable tolerance?
No
Yes
Increase the index by 1
Increase the fluid height by the height increment and place value in the proper index location of the height vector
Calculate the fluid volume at the new height value and place value in the proper index location of the volume vector (use more than one line of code)
Determine number of loop iterations and final fluid volume
Display number of iterations, the safe fill volume and the final fluid volume to the Command Window
Plot the fluid volume vs fluid height and format for technical presentation
Stop

In the RESULTS section of your script, copy and paste as comments the information displayed to the Command Window.

When you are finished with your script, run it and save a *.png file of your figure window. In the figure window, click **File** > **Save As**. Set the file type to *.png. Name the file using the format in the instructions. Submit the plot image to Gradescope along with your m-file.

Instructions

1. Read through the entire problem statement.
2. **With your teammates**: develop and document a plan to solve this problem.
    a. Understand the expectations of the problem.

b. Discuss strategies for solving the problem. This can include citing examples from class notes, drawing pictures, outlining a plan using text or pseudocode, etc. **DO NOT SHARE CODING SOLUTIONS.**

c. Submit your plan to the team assignment in Gradescope

   1. Open the Gradescope assignment **A08 – Team Planning**.
   2. In the area for Problem 4:
      a. Enter the names of your teammates who participated in the planning.
      b. Enter a brief description of your team's plan to solve the problem. The plan should be connected to the problem and have at least 2-3 steps. It should not be a detailed explanation of every step necessary to solve the problem.
      c. If you have image files, etc., that you would prefer to share, then you may add them in the *Optional* file submission area.
   3. Save your results.

d. Add your teammates to the submission (or double-check everyone is added if you completed this step in Problem 3). Select 1 team member to submit the plan. Work together to make sure it is done correctly.

   1. Click **Submit & View Submission** at the bottom of the assignment
   2. Add all teammates to the group (Gradescope instruction link)
   3. All teammates confirm that you get an email confirming the submission and verify that you can see the submission in your Gradescope.

3. **Individually**:

a. Complete your script, run it to get your results, paste the text results as comments into the script and save the figure window as an image.

   - The team plan is an initial start on the problem. It may not be completely correct, and you may find flaws in the plan once you start coding. You should make any individual changes that are necessary to obtain the best solution. You will be assessed on your individual solution to the problem.

b. Cite the teammates you worked with in your script header if their help changed how you decided to solve the problem.

   1. Make sure you also completed the rest of the script header.

c. Submit your properly named files to **A08 – Context Problems** in Gradescope.

Problem 4: Exercise Schedule

Introduction

This problem gives you practice with `for` loops that employ vector indexing. Be sure to follow good programming standards in your script.

Problem

You are planning an exercise schedule with daily exercise minutes. The schedule must follow these guidelines:

- Assign a 30-minute run to every odd numbered day (i.e., every other day).
- Assign a 45-minute run to every even numbered day (i.e., every other day).
- One day every two weeks will have a 90-minute sports day that replaces the scheduled run. The first instance will happen in the first week.
- One day per week will be a rest day (i.e., 0 minutes) that replaces the scheduled run.

You must write a MATLAB script that will create an exercise plan using the guidelines. Your script must do the following:

- Create an Nx7 array that contains the numeric value for number of minutes to exercise each day. The 1st column of the array corresponds to Mondays. N is the number of weeks in the schedule.
- Use only `for` loops to fill out the schedule array. You can use as many `for` loops as you want.
- Ask the user to input information:
  - How many days must be in the exercise schedule?
    - Assume this number will always be divisible by 7 so the schedule does not include an incomplete week.
  - What day of the week will be the sports day (assume Monday is Day 1, Tuesday is Day 2, etc.)?
  - What day of the week will be the rest day (using the same day numbers as above)?
- Follow good programming standards. Initialize the different exercise minutes as variables. Do not hardcode values that are assigned to variables within the script.
- Display to the Command Window the final schedule as an array with each row containing 7 days, where Day 1 is a Monday of the first week, Day 8 is Monday of the second week, etc.
- Make sure to display the label of the days of the week in the correct order along with the numeric array. Hint: Create a string vector with the names of days to display.

Hint: Remember that you can use linear indexing to index array elements in a matrix.

When your script in running properly, run it with the following inputs:

Input item	Input value
Number of days in schedule	35
Day of week for the sports day	4
Day of week for the rest day	5

In the `RESULTS` section of your script, copy and paste as comments the information displayed to the Command Window.

Instructions

1. Read through the entire problem statement.
2. **With your teammates**: develop and document a plan to solve this problem.
    a. Understand the expectations of the problem.
    b. Discuss strategies for solving the problem. This can include citing examples from class notes, drawing pictures, outlining a plan using text or pseudocode, etc. **DO NOT SHARE CODING SOLUTIONS.**
    c. Submit your plan to the team assignment in Gradescope
        1. Open the Gradescope assignment **A08 – Team Planning**.
        2. In the area for Problem 4:
            a. Enter the names of your teammates who participated in the planning.
            b. Enter a brief description of your team's plan to solve the problem. The plan should be connected to the problem and have at least 2-3 steps. It should not be a detailed explanation of every step necessary to solve the problem.
            c. If you have image files, etc., that you would prefer to share, then you may add them in the *Optional* file submission area.
        3. Save your results.
    d. Add your teammates to the submission (or double-check everyone is added if you completed this step in Problem 3). Select 1 team member to submit the plan. Work together to make sure it is done correctly.
        1. Click **Submit & View Submission** at the bottom of the assignment
        2. Add all teammates to the group (Gradescope instruction link)
        3. All teammates confirm that you get an email confirming the submission and verify that you can see the submission in your Gradescope.
3. **Individually**:
    a. Complete your script, run it to get your results, paste the text results as comments into the script and save the figure window as an image.
        - The team plan is an initial start on the problem. It may not be completely correct, and you may find flaws in the plan once you start coding. You should make any individual changes that are necessary to obtain the best solution. You will be assessed on your individual solution to the problem.
    b. Cite the teammates you worked with in your script header if their help changed how you decided to solve the problem.
        1. Make sure you also completed the rest of the script header.
    c. Submit your properly named m-file and data file to **A08 – Context Problems** in Gradescope.

Confirm Your Submission

Problems 1 and 2

You should save your progress on each question in a skills problem so that you do not lose your progress. To confirm your answers, click the **Submit & View Submission** button at the bottom of the questions in Gradescope (or select the assignment name from the Gradescope dashboard, if you have already saved your answers and navigated away from the original submission page).

Confirm that your submission for **A08 – Skills Problems** includes

- □ The function call and instruction text for each skills question;
- □ The expected deliverables and results;
- □ Correct file names for any submitted files, including your Career Account login at the end where required.

You can resubmit your work as many times as you want, but only the final submission will be graded.

Problems 3 and 4 – Individual deliverables

Confirm that your submission for **A08 – Context Problems** includes

- □ The expected deliverables and results;
    - This includes m-files and image files for each problem you complete.
- □ Correct file names for any submitted files, including your Career Account login at the end where required.

You can resubmit your work as many times as you want, but only the final submission will be graded.

Problems 3 and 4 – Team plans

Confirm that your submission for **A08 – Team Planning** includes

- □ The names of all your teammates who participated in each problem's planning;
- □ A brief description of the team's plan for each problem;
- □ All team members included in the group submission.

Assignment Grading

Your work will be graded using the evidences given in the course learning objectives. Familiarize yourself with the LOs and their evidences listed for each problem, which are below. Each non-skill problem's assignment grading has a table and a flowchart. The table outlines what LOs will be used to grade your work and what point values are assigned to each evidence. The flowchart outlines the grading process that a grader will use to assess your work.

Find the list of the course LOs, with evidences, on Brightspace (Content > Key Course Info > Learning Objectives).

Team Planning

Each problem's plan is worth 1 point.

Individual Problems

Problem 1

LOs: PC05, MAT06

Problem 1 is worth 5 points. There is partial credit. The partial credit may be more specific than what is in the course LOs and is based on evidences in MAT06.

You must meet the PC05 expectations for each question. If you do not meet these, you will lose additional credit.

Evidence	Penalty
PC05 (1)	Lose full credit on problem
PC05 (2)	Lose 25% of full credit on problem
PC05 (3)	Lose 25% of full credit on problem
PC05 (4)	Lose 5% of full credit on problem

Problem 2

LOs: PC05, MAT06 and MAT07

Problem 2 is worth 5 points. There is partial credit. The partial credit may be more specific than what is in the course LOs and is based on evidences in MAT06 and MAT07.

You must meet the PC05 expectations for this problem. If you do not meet these, you will lose additional credit.

Evidence	Penalty
PC05 (1)	Lose full credit on problem
PC05 (2)	Lose 25% of full credit on problem
PC05 (3)	Lose 25% of full credit on problem
PC05 (4)	Lose 5% of full credit on problem

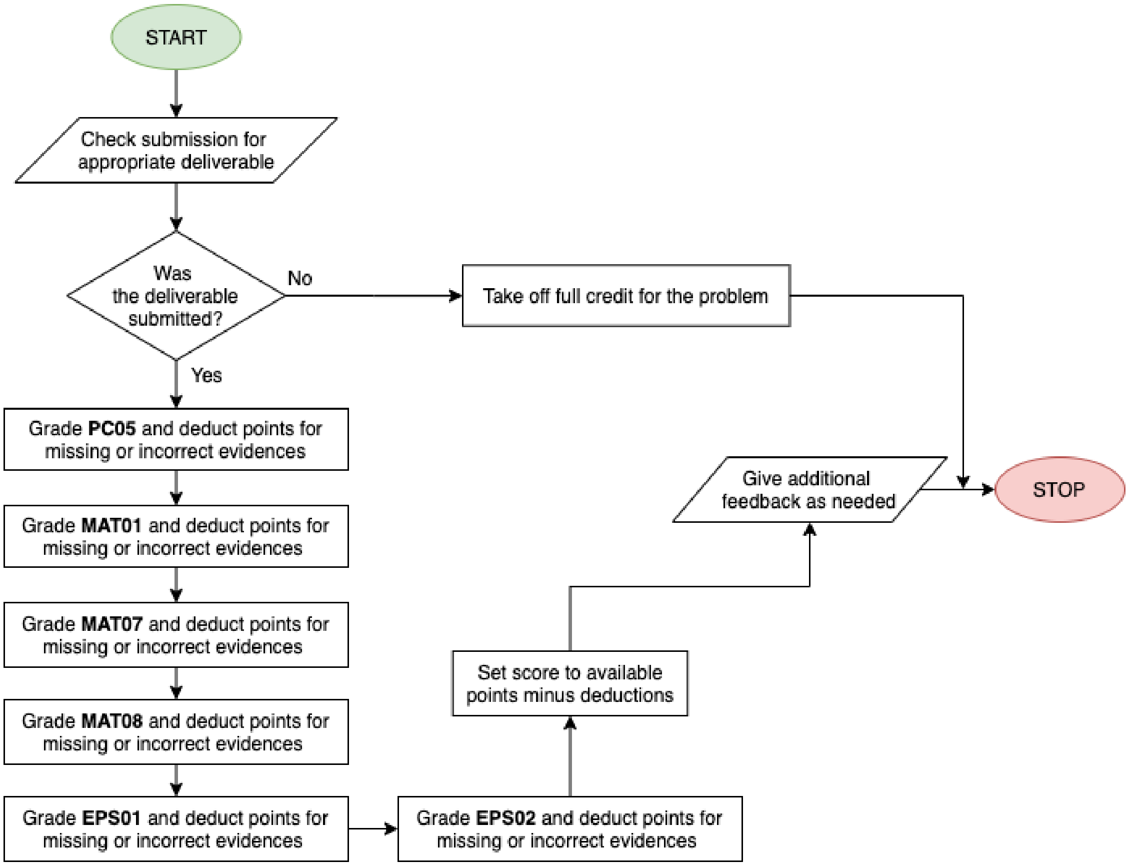
Problem 3

LO Table

Note: PC05 evidences are only deductions since you are expected to follow the assignment instructions.

	PC05	MAT01	MAT07	MAT08	EPS01	EPS02
(1)	-5	0.3	0.4	0	0	0
(2)	-0.5	0	0.4	0.6	0	0.2
(3)	-1.25	0	0.4	0.3	0.2	0.2
(4)	-0.25	0	0.4	0	0.2	0
(5)	0	0.2	0.4	0	0	0
(6)	0	0	0.4	0	0	0
(7)	0	0.2	0	0	0	0.2
(8)	0	0	0	0	0	0

Grading Process


START
Check submission for appropriate deliverable
Was the deliverable submitted?
No
Take off full credit for the problem
Yes
Grade **PC05** and deduct points for missing or incorrect evidences
Grade **MAT01** and deduct points for missing or incorrect evidences
Grade **MAT07** and deduct points for missing or incorrect evidences
Grade **MAT08** and deduct points for missing or incorrect evidences
Grade **EPS01** and deduct points for missing or incorrect evidences
Grade **EPS02** and deduct points for missing or incorrect evidences
Set score to available points minus deductions
Give additional feedback as needed
STOP

Problem 4

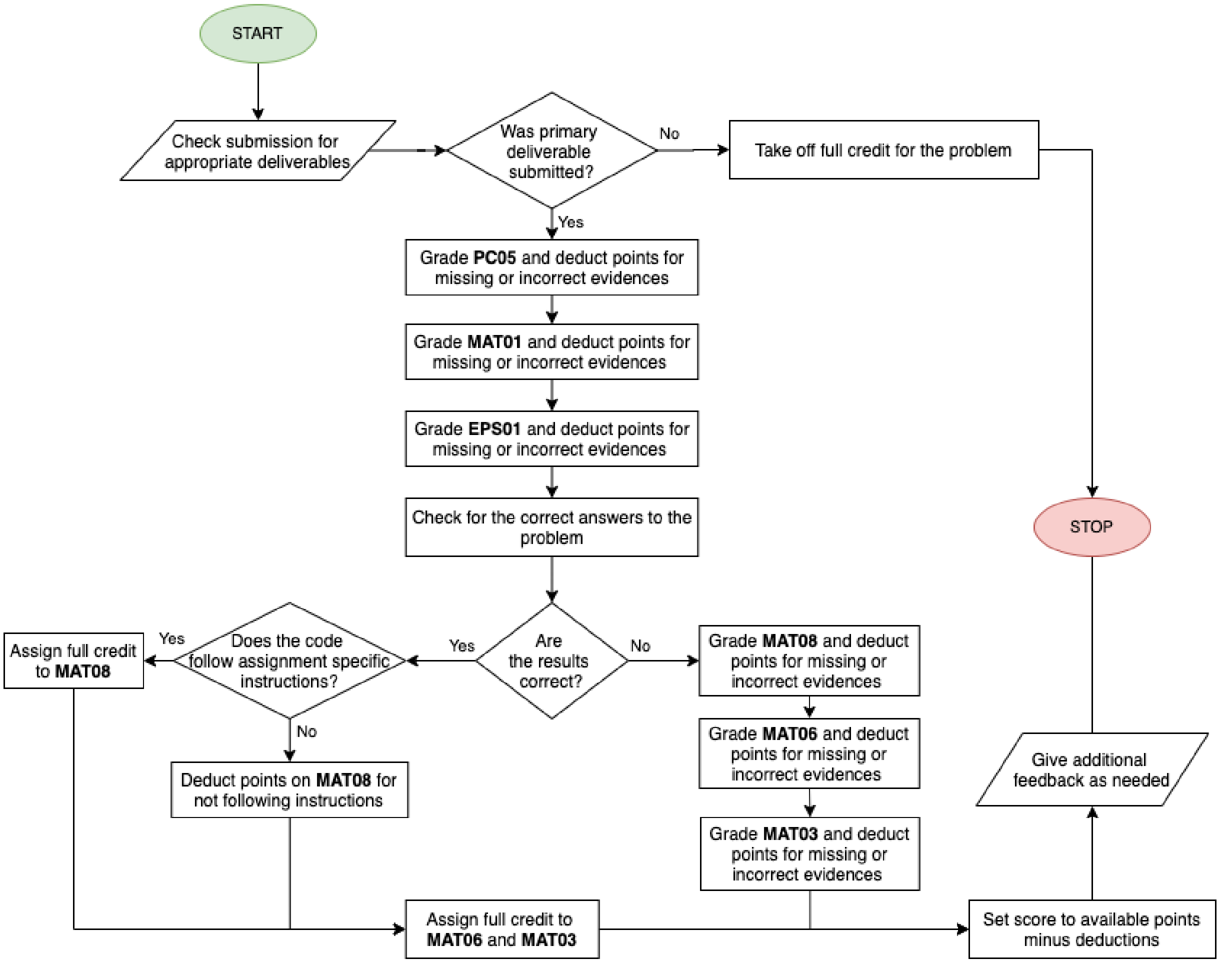
LO Table

This table lists every LO evidence that will be assessed and how many points will be given for that evidence. Refer the individual evidences in the learning objectives for more explicit details.

Note: PC05 evidences are only deductions since you are expected to follow the assignment instructions.

	PC05	MAT01	EPS01	MAT08	MAT06	MAT03
(1)	-5	0.1	0	0.6	0.5	0.2
(2)	0	0	0	0.5	0.5	0
(3)	-1.25	0.1	0	0.3	0.5	0
(4)	-0.25	0.1	0	0	0.5	0
(5)	0	0.1	0.3	0	0.5	0
(6)	0	0.1	0	0	0	0
(7)	0	0.1	0	0	0	0
(8)	0	0	0	0	0	0

Grading Process


START
Check submission for appropriate deliverables
Was primary deliverable submitted?
No
Take off full credit for the problem
Yes
Grade **PC05** and deduct points for missing or incorrect evidences
Grade **MAT01** and deduct points for missing or incorrect evidences
Grade **EPS01** and deduct points for missing or incorrect evidences
Check for the correct answers to the problem
Are the results correct?
Yes
Does the code follow assignment specific instructions?
Yes
Assign full credit to **MAT08**
No
Deduct points on **MAT08** for not following instructions
No
Grade **MAT08** and deduct points for missing or incorrect evidences
Grade **MAT06** and deduct points for missing or incorrect evidences
Grade **MAT03** and deduct points for missing or incorrect evidences
Assign full credit to **MAT06** and **MAT03**
Set score to available points minus deductions
Give additional feedback as needed
STOP