A06 · Logic and Selection Structures

Introduction

Assignment Goals

This assignment focuses on two key programming tools in MATLAB: logical statements and selection structures. You will gain programming experience with MATLAB logic statements and then you will apply those skills to an engineering context. You will do the same for selection structures, which require logic statements as conditions.

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 **A06 – Skills Problems**		
Problem	**Type**	**Deliverables**
Problem 1: MATLAB Skills - Logic Statement	Individual	□ Requested code snippets and information
Problem 2: MATLAB Skills - Selection Structures	Individual	□ Requested code snippets and information
Submit Problems 3 and 4 to the Gradescope programming assignment **A06 – Context Problems**		
Problem	**Type**	**Deliverables**
Problem 3: Landslide Monitoring	Individual	□ A06Prob3_landslide_*login*.m □ Supporting file(s): ○ Data file loaded into m-file
Problem 4: Airline Seat Assignment	Individual	□ A06Prob4_seating_*login*.m □ Supporting file(s): ○ Data file loaded into m-file

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 logical operations; problems 2 and 4 require selection structures. 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. 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 11**. 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:

`min, max, mean, find, any, all, if, elseif, else, end, error, input`

`find` Command in the MATLAB Editor

If you use the `find` command and edit your code within the MATLAB editor, you may notice that MATLAB sometimes produces a warning on the lines with the `find` command. MATLAB may suggest that you use logical indexing, which allows you to use logical 1s and 0s to identify which values in a vector correspond to the 'true' condition, similar to array indexing. You can use either method, `find` or logical indexing, when appropriate. There may be times where only one of the them works. You should know how the `find` command works for exams.

Intervals and Ranges

A range of numbers can be defined as an interval. If a boundary value is included within the range, then we say the range is "inclusive" of the number. If a boundary value is not included in the range, then the range is "exclusive" (or not inclusive) of that number. See examples below for the range of number 5 to 10.

Relational notation	Interval notation	Explanation
$5 \leq x \leq 10$	$[5, 10]$	The numbers between 5 and 10, inclusive of both 5 and 10
$5 \leq x < 10$	$[5, 10)$	The numbers between 5 and 10, inclusive of only 5 (exclusive of 10)
$5 < x \leq 10$	$(5, 10]$	The numbers between 5 and 10, inclusive of only 10 (exclusive of 5)
$5 \leq x \leq 10$	$(5, 10)$	The numbers between 5 and 10, exclusive of both 5 and 10

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:

- **A06 – Skills Problems**: submit your deliverables for Problems 1 and 2
- **A06 – Context Problems**: submit your deliverables for Problems 3 and 4
- **A06 – 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 **A06_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 – Logic Statements

Introduction

This problem allows you to practice writing logical statements using built-in functions and relational and logical operations on a matrix. You will submit your answers to an online assignment on Gradescope.

Problem

This problem has 5 parts. Make sure **A06_skills.p** is in your current MATLAB folder (see the *Notes Before You Start* section above).

General Instructions

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

   ```
   >> A06_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. Do not use any looping or selection structures. You can and should complete each question using logic statements and built-in functions.
      i. Each question can be answered with a single line of code, but you are allowed to use more than one command as long as they are in the proper order.

   b. You may hardcode any values given in the question text, but do not hardcode values of `mat1` into your commands.

   c. Assign all calculations to variables. Make sure the variables are valid in MATLAB, but you are not required to use detailed variable names as described in the programming standards. Variables like `partA` are acceptable.

   d. Your commands must run in MATLAB to receive full credit.

   e. Programming standards will not be graded. Do not include comments in your solutions.
3. Submit your work in Gradescope:
   a. Open Gradescope > **A06 – 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 solution(s) for each part. 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 – Selection Structures

Introduction

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

Problem

This problem presents a selection structure in language format. You will convert the language to code.

Instructions

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

   ```
   >> A06_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.
   - Keep the conditions in the order they are presented in the instruction text.
   - You can test your structure with any integer for A. You should attempt to test each path of the structure.
     - It is possible that your instruction text may require a path that is never used. If this happens, code the conditions as given in the instruction text.
   - Programming standards will not be assessed in this problem. Do not include comments in your solutions.
3. Submit your work in Gradescope:
   a. Open Gradescope > **A06 – Skills Problems** 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 solution(s). Follow any additional instructions.

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

Problem 3: Landslide Monitoring

Introduction

This problem focuses on your ability to write a MATLAB script that uses relational and logical operations to answer engineering-based questions. You will also use programming commands to display text and numeric results to the Command Window in a professional manner. Be sure to follow good programming standards in your script.

Problem

A civil engineer has been tasked with sorting and understanding landslide data from around the world. You will practice using relational and logical operators with the data file named **Data_global_landslide.csv** to help the civil engineer answer some important questions about the data.

The datafile contains 8 columns of data. There are three categories of landslides in the columns labelled "landslide category", each with a category number. The same is true for the columns "landslide trigger", which contains numbers representing what triggered each landslide, and "landslide size" which tells us whether the landslide was small, medium or large. The table below explains what the different numbers mean for each column:

Landslide category	Landslide trigger	Landslide size
1: landslide	10: rain	100: small
2: mudslide	20: continuous rain	200: medium
3: snow avalanche	30: downpour	300: large
	40: snow fall or snow melt	

Import the data file into MATLAB using commands within your script and use it to answer the following questions.

1. How many landslides occurred that were categorized as "mudslide" whose landslide trigger was not "downpour", or landslides that were triggered by "downpour" but were not categorized as "mudslide"?
2. What was the average latitude of the landslides that were categorized as "snow avalanche"?
3. How many cases involved any type of landslide with fatalities greater than 50 that occurred within latitudes greater than 0 or longitudes less than 0 but not both?
4. How many "mudslides" and "landslides" occurred due to "snow fall or snow melt" that had landslide sizes that were at least "medium" with latitudes above the equator? (the equator has a latitude of 0)
5. How many and in which year(s) were there "medium" sized landslide categories of "snow avalanche" in the USA? You know that the latitude for USA should range between approximately 19.5 to 64.86 while the longitude should be negative.

Display your results using professional and easily read format. *Display Guidelines:*

- Print values within an appropriate sentence that answers the question.
- Use appropriate numerical formatting (e.g., control the number of decimal places displayed.)
- Identify the question number in the text display.

NOTE: Do not use any loops or other non-sequential structures to answer these questions.

In the `RESULTS` section of the template, paste your text display for each question as comments.

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 **A06 – Team Planning**.
  2. In the area for Problem 3:
     - 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 (even if Problem 4 is not finished). 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.

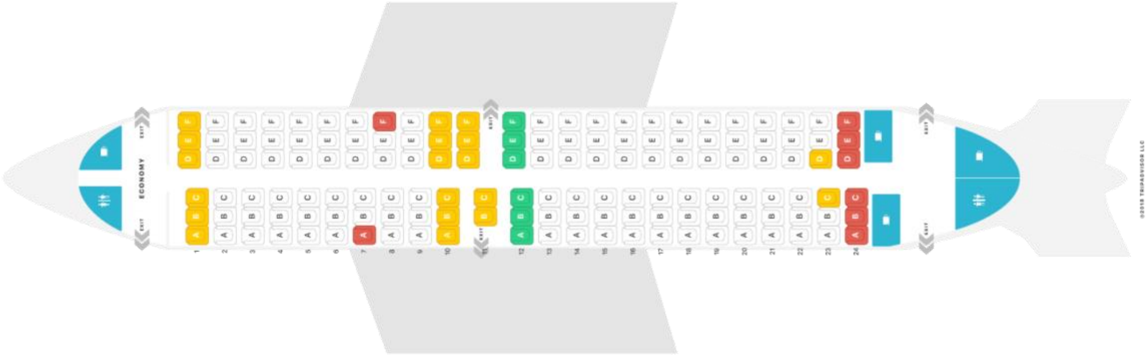
3. **Individually**:
   - a. Complete your script, run it to get your results, and paste those results as comments into the script.
     - o The team plan is an initial start on the problem. It may not be completely correct, and you may find flaws in the plan once you start coding. You should make any 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 **A06 – Context Problems** in Gradescope.

Problem 4: Airline seat assignment

Introduction

This problem focuses on your ability to translate an engineering-context flowchart, which has a selection structure, into a MATLAB script, and then use the script to test the selection structure. Be sure to follow good programming standards in your script.

Problem

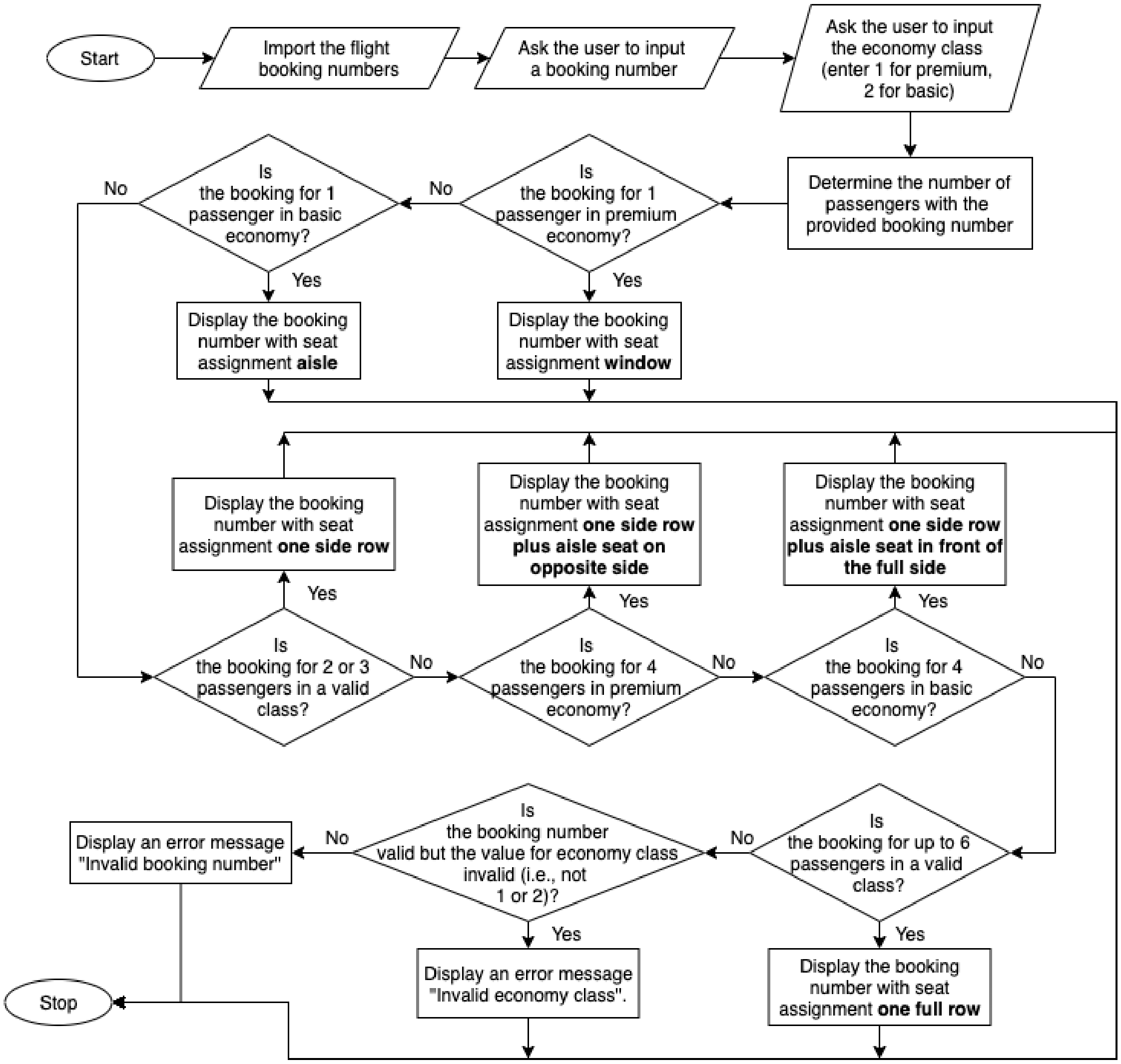


Airlines have a system that assigns seats to passengers in a flight. An airline has set new health requirements to ensure socially distant seating for passengers from different households. The airline wants you to help them write code to assign seat locations based on the number of passengers in one booking and their booking class (premium economy or basic economy).

The aircraft seats 6 people in one row, with 3 on one side of the aisle and 3 on the other. The airline has set these options for assigning seats: window, aisle, one side row, one side row plus aisle seat on opposite side, one side row plus aisle seat in front of the full side, or full row (all six seats, across the aisle). The seat location may depend on the number of passengers per booking number and the booking class (see the flowchart below).

You can assume that passengers with the same booking number are from the same household and can sit next to each other. The airline allows a maximum of 6 people per booking. Note: the airline does not want you to assign a seat number, only determine the corresponding seat option according to the flowchart.

You have been provided a data file, named **Data_booking_numbers.txt**, that contains the booking numbers for all passengers on an upcoming flight. The booking could be in premium economy class or basic economy class. For this code, assume the user knows the class and can enter it when requested. The flowchart below shows how the airline wants you to assign seats to the passengers. Translate the flowchart to a script.


Start
Import the flight booking numbers
Ask the user to input a booking number
Ask the user to input the economy class (enter 1 for premium, 2 for basic)
Determine the number of passengers with the provided booking number
Is the booking for 1 passenger in premium economy?
Yes
Display the booking number with seat assignment **window**
No
Is the booking for 1 passenger in basic economy?
Yes
Display the booking number with seat assignment **aisle**
No
Is the booking for 2 or 3 passengers in a valid class?
Yes
Display the booking number with seat assignment **one side row**
No
Is the booking for 4 passengers in premium economy?
Yes
Display the booking number with seat assignment **one side row plus aisle seat on opposite side**
No
Is the booking for 4 passengers in basic economy?
Yes
Display the booking number with seat assignment **one side row plus aisle seat in front of the full side**
No
Is the booking for up to 6 passengers in a valid class?
Yes
Display the booking number with seat assignment **one full row**
No
Is the booking number valid but the value for economy class invalid (i.e., not 1 or 2)?
Yes
Display an error message "Invalid economy class".
No
Display an error message "Invalid booking number"
Stop

Once you have your code written, run it with the following test cases:

Test Case	Booking Number	Economy Class Number
1	13845360	2
2	17248113	1
3	17657472	2
4	18276759	2
5	12774978	1
6	21004657	2
7	11174805	1
8	13925866	1

For each test case, paste your printed display into the RESULTS section of your script as comments.

After you finish your test cases, answer this question in the ANALYSIS section of your script. Write your answer as MATLAB comments.

Q1. Does each condition in the flowchart get tested at least one time with the test cases listed above? If no, then list all conditions that are **not** tested. Format your answer following the guidelines in EPS01.

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 **A06 – 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 for each test case to get your results, and paste those results as comments into the script. Answer the analysis question.
        - 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 **A06 – 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 **A06 – 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 **A06 – Context Problems** includes

- □ The expected deliverables and results;
  - ○ This includes m-files and data 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 **A06 – 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.

References

https://data.nasa.gov/Earth-Science/Global-Landslide-Catalog-Export/dd9e-wu2v

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, MAT03

Problem 1 is worth 5 points, where each part is 1 point. There is some partial credit on parts B-E. The partial credit may be more specific than what is in the course LOs and is based on evidence MAT03 (1) and (5).

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 (8)	Lose 10% of full credit on problem

Problem 2

LOs: PC05, MAT06

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.

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

Problem 3

LO Table:

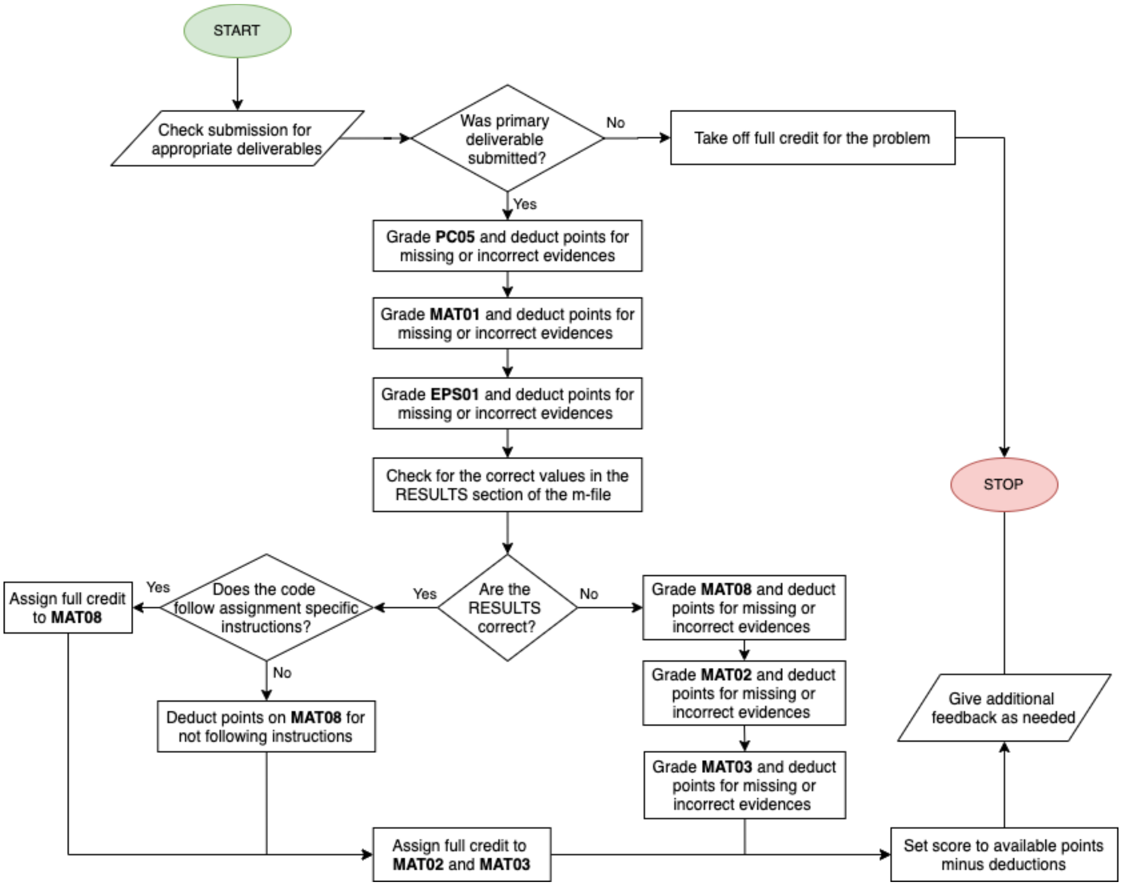
This table lists every LO evidence that will be assessed and how many points will be given for that evidence. Refer the individual evidences in the learning objectives for more explicit details.

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

	PC05	MAT01	EPS01	MAT08	MAT02	MAT03
(1)	-1.25	0.1	0.0	0.25	0.2	0.3
(2)	-0.25	0.1	0.0	0.5	0.2	0.1
(3)	-0.5	0.1	0.4	0.25	0.2	0.1
(4)	0	0.2	0.3	0	0	0.1
(5)	0	0.1	0.4	0	0	0.4
(6)	0	0.2	0	0	0	0
(7)	0	0.1	0	0	0	0
(8)	0	0.4	0	0	0	0

Grading Process:

This flowchart outlines how the grader will assess your work for this problem.


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 values in the RESULTS section of the m-file
STOP
Are the RESULTS correct?
Yes
No
Does the code follow assignment specific instructions?
Yes
Assign full credit to **MAT08**
No
Deduct points on **MAT08** for not following instructions
Grade **MAT08** and deduct points for missing or incorrect evidences
Grade **MAT02** and deduct points for missing or incorrect evidences
Grade **MAT03** and deduct points for missing or incorrect evidences
Give additional feedback as needed
Assign full credit to **MAT02** and **MAT03**
Set score to available points minus deductions

Problem 4

LO Table

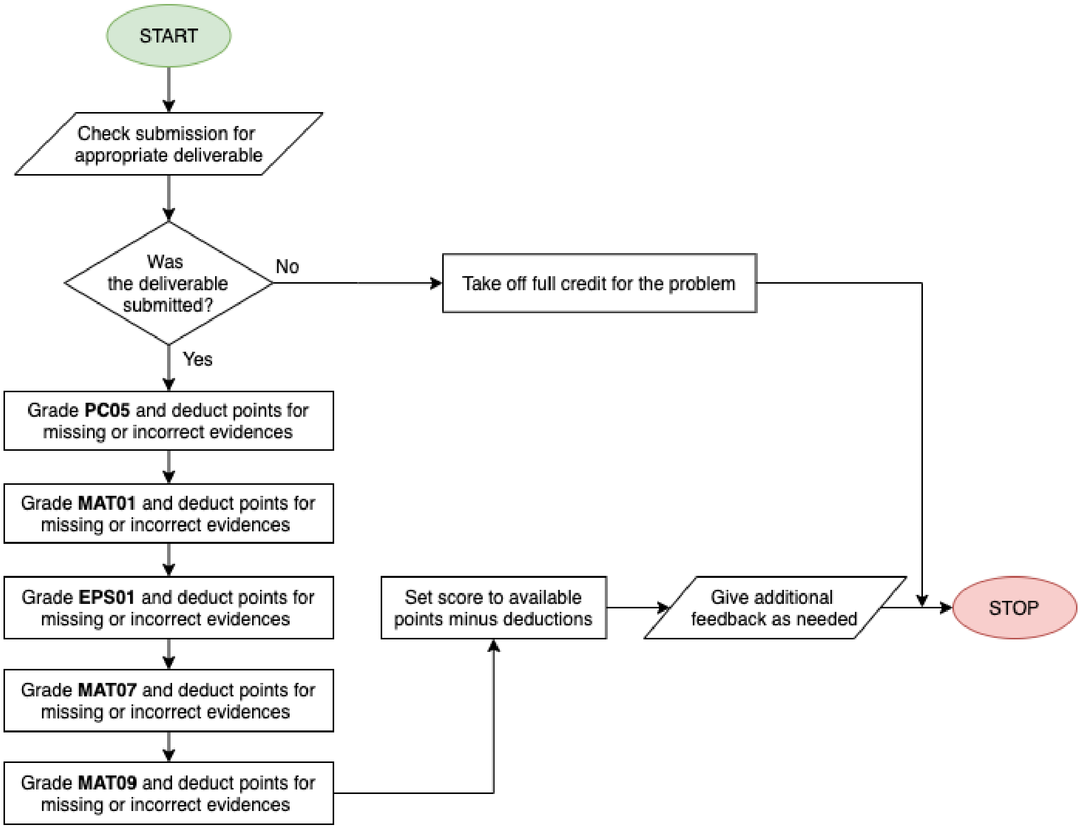
This table lists every LO evidence that will be assessed and how many points will be given for that evidence. Refer the individual evidences in the learning objectives for more explicit details.

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

	PC05	MAT01	EPS01	MAT07	MAT09
(1)	-1.25	0.2	0	0.3	0.3
(2)	-0.25	0.1	0	0.3	0.6
(3)	-0.5	0.1	0	0.3	0.6
(4)	0	0.1	0	0.3	0
(5)	0	0.3	0.3	0.3	0
(6)	0	0.2	0	0.3	0
(7)	0	0.2	0	0	0
(8)	0	0.2	0	0	0

Grading Process

This flowchart outlines how the grader will assess your work for this problem.


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 **EPS01** and deduct points for missing or incorrect evidences
Grade **MAT07** and deduct points for missing or incorrect evidences
Grade **MAT09** and deduct points for missing or incorrect evidences
Set score to available points minus deductions
Give additional feedback as needed
STOP