

A09 · Repetition Structures and User-Defined Functions

Introduction

Assignment Goals

This assignment focuses on two programming tools: definite repetition structures (for loops) and user-defined functions. You will learn to interpret a for loop flowchart and to code for loops to achieve an outcome for a context. You will gain experience creating user-defined functions and using nested structures to check inputs of UDFs.

Successful Completion

This assignment has four (4) problems. Problems 1 and 2 go with Class A; Problems 3 and 4 go with Class B.

1. Read *Notes Before You Start*, on **Page 1**.
2. Read each problem carefully. You are responsible for following all instructions within each problem.
 - a. Problems 1 and 2 require FOR loops. Problems 3 and 4 require user-defined functions (UDFs).
 - b. The deliverables list within each problem contains everything you are expected to submit.
 - c. You will need the problem generator **A09_skills.p** for Problems 1, 3, 4. See [this link](#) to view instructions.
3. Complete the problems using the problem-specific m-file templates when a template is provided in the assignment download.
4. For any file, replace *template* or *login* in the filename with your Purdue Career Account login.
5. Review your work using the learning objective evidences.
6. When your work is complete, confirm your deliverables are submitted to Gradescope.
 - a. Note the two different assignments in Gradescope. See *Notes Before You Start*.
 - i. Submit *individual work* for all problems (skill and context) to **A09 – All Problems**.
 - ii. Submit *team plan* for Problem 2 to **A09 – Team Planning**.
 - b. You can resubmit your work as many times as you want; only the final submission will be graded.
 - c. Do **NOT** upload any document not listed in the deliverables. Do not upload temporary versions of m-files (*.m~ or *.asv) – these files will be ignored by Gradescope.
7. Late submissions will be accepted up to 24 hours after the due date and will result in a 25% penalty.

Learning Objectives & Grading

This course uses learning objectives (LOs) to assess your work. You can find a full list of the course LOs [here](#). Review the grading outline at the end of each problem in this assignment to see each problem's LOs.

Notes Before You Start

Helpful MATLAB Commands

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

`for`, `end`, `function`, `error`, `rem`

String Arrays in MATLAB

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 array commands, such as [length](#) and [size](#), on string arrays.

Gradescope

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

- **A09 – All Problems:** submit your deliverables for all problems in this assignment ([Help link](#))
- **A09 – Team Planning:** submit your team plan for Problem 2 ([Help link](#))

Problem 1: 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.

Problem Generator Information

If you have questions about how to use the problem generator, review [this link](#) that shows step-by-step instructions.

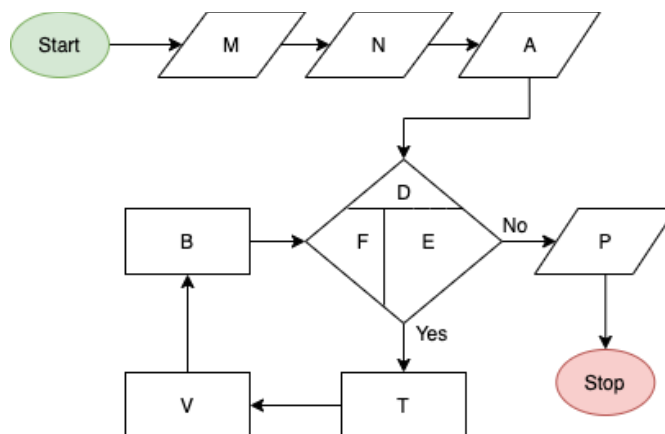
File Name	PUID	Problem Number
A09_skills	Your 8-digit PUID	1

Submission

Gradescope Assignment	A09 – All Problems	Assignment Type	Individual
Deliverables	<input type="checkbox"/> Requested information <input type="checkbox"/> A09Prob1_for_login.m		

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 the problem generator. 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. Enter the problem generator function call into the MATLAB Command Window prompt.
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 instruction text with run receipt and m-file into Gradescope.
 4. Save your answers in Gradescope. See [this video](#) for help (this is an untimed assignment).

Grading

LOs: PC05, MAT03, MAT06

Point value: 7 points. The partial credit may be more specific than what is in the course LOs and is based on evidence MAT03 (5) and MAT06. If you do not meet the PC05 expectations, 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 15% of full credit on problem

Problem 2: 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.

Submission

Individual

Gradescope Assignment	A09 – All Problems	Assignment Type	Individual
Deliverables	<input type="checkbox"/> A09Prob2_exercise_login.m		

Team Plan

Gradescope Assignment	A09 – Team Planning	Assignment Type	Team
Deliverables	<input type="checkbox"/> Requested information		

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 activity and is not the same as the sports day.

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

- Ask the user to input information:
 - How many days must be in the exercise schedule? The schedule must include full weeks (7 days per 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)?
 - Warn and then re-ask the user for information if the number of days in the schedule is not divisible by 7 or if the sports day and rest day are entered as the same day. The code should repeat this process until all three numbers are valid.
- Create an $N \times 7$ 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.
- 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 is running properly, run it with the following inputs:

Input item	Input value
Number of days in schedule	49
Day of week for the sports day	3
Day of week for the rest day	6

In the **RESULTS** section of your script, copy and paste as comments the text 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 for this assignment's team plan (see the submission list at the beginning of this problem).
 2. In the area for this problem:
 - 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. 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 a submission email and verify that you can see the submission in your Gradescope.
3. **Individually:**
 - a. Complete your script, run it to get your results, and add the results to the m-file as comments.
 - 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 any peers you worked with in your script header if their help changed how you decided to solve the problem. Make sure you also completed the rest of the script header.
 - c. Submit your properly named m-file to the appropriate problem in the individual Gradescope assignment. Do not submit any other files.

Grading

LOs: PC05, MAT01, EPS01, MAT07, MAT09

Team plan: 1 point

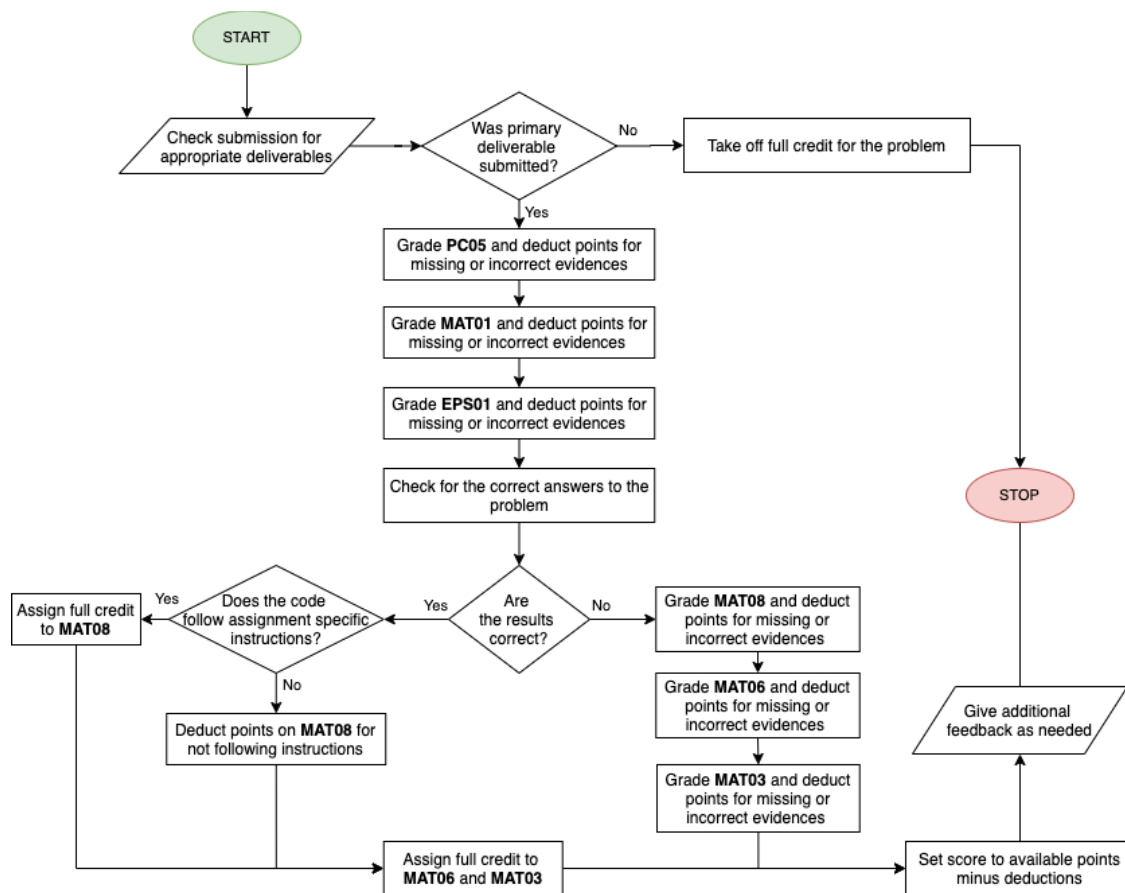
Individual assignment point value: 9 points. Partial credit is possible; see the LO table below for details. You must meet the PC05 expectations. If you do not meet these, you will lose additional credit.

LO Table

	PC05	MAT01	EPS01	MAT08	MAT06	MAT03
(1)	-100%	0.3	0	1.4	0.7	0.5
(2)	25%	0	0	0.7	0.7	0
(3)	0	0.2	0	0.4	0.7	0
(4)	-15%	0.3	0.4	0	0.7	0
(5)	0	0.2	0.3	0	0.7	0
(6)	0	0.3	0	0	0	0
(7)	0	0.2	0	0	0	0
(8)	0	0.3	0	0	0	0

Grading Process

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



Problem 3: MATLAB Skills – UDFs

Introduction

This problem requires you to write and run a user-defined function (UDF).

Problem Generator Information

If you have questions about how to use the problem generator, review [this link](#) that shows step-by-step instructions.

File Name	PUID	Problem Number
A09_skills	Your 8-digit PUID	3

Submission

Gradescope Assignment	A09 – All Problems	Assignment Type	Individual
Deliverables	<input type="checkbox"/> Requested information and your appropriately named m-file		

Problem

Instructions

1. Enter the problem generator function call into the MATLAB Command Window prompt.
2. Read the instructions that display to the Command Window. There are four (4) parts.
 - Use the included template to write the requested function for Part A. Rename the template with your function name defined in Part A.
 - Run your function to answer Parts B and C.
 - Answer Part D **without** changing your function m-file from Part A. Enter only the function definition line in the box in Gradescope.
 - Programming standards will not be assessed in this problem. Do not include comments in your solutions. Properly name the file.
3. Submit your instruction text with run receipt and m-file into Gradescope.
4. Save your answers in Gradescope. See [this video](#) for help (this is an untimed assignment).

Grading

[LOs](#): PC05, MAT05

Point value: 7 points. The partial credit may be more specific than what is in the course LOs and is based on evidence MAT05. If you do not meet the PC05 expectations, 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 15% of full credit on problem

Problem 4: MATLAB Skills – UDF and Nested Structures

Introduction

This problem requires you to read a flowchart that shows the coding plan for a user-defined function that contains a nested structure (where the flowchart contains at least one selection structure).

Problem Generator Information

If you have questions about how to use the problem generator, review [this link](#) that shows step-by-step instructions.

File Name	PUID	Problem Number
A09_skills	Your 8-digit PUID	4

Submission

Gradescope Assignment	A09 – All Problems	Assignment Type	Individual
Deliverables	<input type="checkbox"/> Requested information, image file, and your appropriately named m-file		

Problem

1. Enter the problem generator function call into the MATLAB Command Window prompt.
2. Read the flowchart in the figure window and the instructions that display to the Command Window.
 - You will need internet access to run this problem. The flowchart image is stored on a purdue.engineering.edu server. If this is a problem for you, contact your GTA and instructor for help.
 - Use the included template to write the requested function. Rename the template with your function name defined in the instruction text.
 - Translate the flowchart exactly as written, even if there are ways to achieve the same result.
 - The instructions will tell you if `out1` is expected to be a scalar or a vector.
 - Remember: if you divide an integer by 1, you will get a remainder of 0.
 - Programming standards will not be assessed in this problem. Use the variable names used in the instruction text. Do not include comments in your solutions. Properly name your m-file.
 - Save an image file of your flowchart figure (see [help](#) if needed).
3. Submit your instruction text with run receipt, instruction flowchart image, and m-file into Gradescope.
4. Save your answers in Gradescope. See [this video](#) for help (this is an untimed assignment).

Grading

[LOs](#): PC05, MAT05, MAT06

Point value: 7 points. The partial credit may be more specific than what is in the course LOs and is based on evidence MAT05. If you do not meet the PC05 expectations, 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 15% of full credit on problem