A11 · Complex Loops

Introduction

Assignment Goals

Two skills questions will give you practice working with complex looping structures. You will then work with complex loops using engineering context.

Successful Completion

This assignment has **3** problems. All problems go with Classes A and 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. The deliverables list within each problem contains everything you are expected to submit.
    b. You will need the problem generator **A11_skills.p** for Problems 1 and 2. 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 three different assignments in Gradescope.
        i. **A11 – Skills Problems**: submit your deliverables for Problems 1 and 2. Help link.
        ii. **A11 – Context Problems**: submit your individual deliverables for Problem 3. Help link.
        iii. **A11 – Team Planning**: submit your team plan for Problem 3 as a team. Help link.
    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:

`end` (as an index value in an array)

Problem 1: MATLAB Skills – Removing Loops

Introduction

This problem allows you to practice removing a `for` loop with vector indexing from code and replacing it with appropriate vector operations. You will also use skills from relational and logical operations to identify elements for the calculations.

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
A11_skills	Your 8-digit PUID	1

Submission

Gradescope Assignment	A11 – Skills Problems	Assignment Type	Individual
Deliverables	☐ Requested results and information ☐ A11Prob1_noLoop_*login*.m		

Problem

The instruction text will provide you with the code for a `for` loop with vector indexing. This code assigns a vector of values, and then the loop moves through each element in the vector. Depending on the value of the element, the code will perform one of two possible calculations. The calculation is then added in the index location of the final results vector, named `newvec`.

You must write a script that starts with the same original vector variable, `vec`, and will produce the same final vector as `newvec`, **without** using loops (for loops or while loops) and without using selection structures. Name the final vector of your no-loop code `newvec_noLoop`. Submit your script to Gradescope.

Instructions

1. Enter the problem generator function call into the MATLAB Command Window prompt.
2. Read the written instruction text that appears in the Command Window. Write a script to complete the task.
    - Do not include the original code in your script. You will paste that code into the instruction text box in Gradescope as reference.
    - Initialize `vec` in your script with the vector you were given in the instruction text.
    - Your script must work for any vector `vec`, not just the one you were given (i.e., avoid hardcoding your solution for your specific values in `vec`).
    - Use the included template to write the requested code.
    - Programming standards will not be assessed in this problem. Do not include comments in your solutions. Properly name the file.
2. Submit your instruction text with run receipt and m-file into Gradescope.
3. Save your answers in Gradescope. See this video for help (this is an untimed assignment).

Grading

LOs: PC05, MAT06

Point value: 12 points. The partial credit may be more specific than what is in the course LOs and is based on evidences in 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 10% of full credit on problem
PC05 (4)	Lose 15% of full credit on problem

Problem 2: MATLAB Skills – Nested Loops

Introduction

This problem requires you to read a nested-loop flowchart and translate it into a script.

Problem Generator Information

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

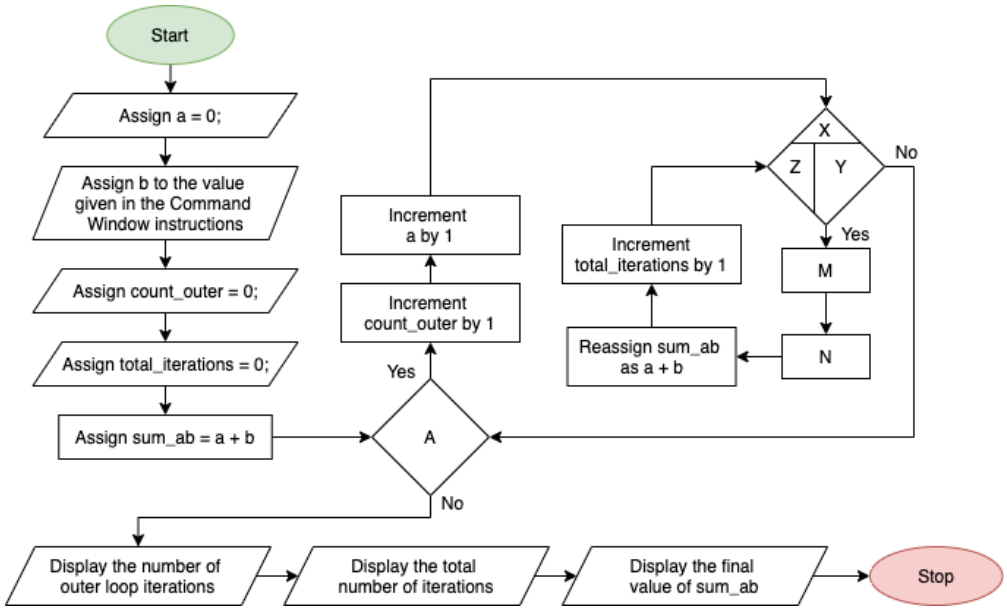
File Name	PUID	Problem Number
A11_skills	Your 8-digit PUID	2

Submission

Gradescope Assignment	A11 – Skills Problems	Assignment Type	Individual
Deliverables	☐ Requested results and information ☐ A11Prob2_nested_*login*.m		

Problem

You have this partially completed flowchart:



To get the remaining instructions for this flowchart, you will need to use the problem generator. Match the instruction 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 written instruction text that appears in the Command Window. Write a script to complete the task.
    - 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.
- Use the variable names stated in the flowchart.
- Submit your script **and** displayed results to Gradescope.
- Programming standards will not be assessed in this problem. Properly name your script.

3. Submit your instruction text with run receipt, results, and m-file into Gradescope.
4. Save your answers in Gradescope. See this video for help (this is an untimed assignment).

Grading

LOs: PC05, MAT06 and MAT07

Point value: 8 points. The partial credit may be more specific than what is in the course LOs and is based on evidences in MAT06 and MAT07. 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 3: Pixel Filtering

Introduction

You will apply your complex looping skills to an engineering context.

Submission

Individual

Gradescope Assignment	A11 – Context Problems	Assignment Type	Individual
Deliverables	☐ A11Prob3_pixelFilters_*login*.m Supporting file(s): ☐ A11Prob3_figure_*login*.png ☐ Data_pixels.csv		

Team Plan

Gradescope Assignment	A11 – Team Planning	Assignment Type	Team
Deliverables	☐ Requested information		

Problem

Image and signal arrays can require digital filtering. One filtering method is to identify an element in the array, find the median of it and its surrounding elements, and then replace the element with the median value. For example:

Replace the element in the second row, second column with the median of it and the 8 values surrounding it:

1	8	1	3	4
7	9	8	6	2
5	10	6	0	1
2	8	9	4	6
0	2	8	6	6

$$y_{2,2} = \text{median}(1, 8, 1, 7, 9, 8, 5, 10, 6)$$

$$y_{2,2} = 7$$

1	8	1	3	4
7	7	8	6	2
5	10	6	0	1
2	8	9	4	6
0	2	8	6	6

Then repeat this for the next element in the array. For this example, we will move one element to the right (to the next horizontal element):

1	8	1	3	4
7	7	8	6	2
5	10	6	0	1
2	8	9	4	6
0	2	8	6	6

$$y_{2,3} = \text{median}(8, 1, 3, 7, 8, 6\ 10, 6, 0)$$

$$y_{2,3} = 6$$

1	8	1	3	4
7	7	6	6	2
5	10	6	0	1
2	8	9	4	6
0	2	8	6	6

Notice that the calculation used the median value of $y_{2,2}$ calculated in example 1 to find the new value for $y_{2,3}$.

Continue this process until all elements inside the border rows and columns have had the filter applied. This type of filter can start at any corner of the matrix but must start one row and one column from the border (at $y_{2,2}, y_{4,2}, y_{2,4}$, or $y_{4,4}$ in the example matrix). This filter cannot change any element that is on the border rows or columns of the array.

Using the same type of median filter described, create a function that meets the following criteria:

- Accepts no input arguments or output arguments
- Imports the pixel values from **Data_pixels.csv** using built-in function. Do not hardcode the data values.
- Using the imported data, applies three different median filters using three different starting points. All filters must start from the original data matrix. Do not overwrite the original data array.

- Filter 1: start at index (2,2), move left to right from top to bottom, and end at index (last row - 1, last column - 1)
- Filter 2: start at index (last row - 1, last column - 1), move right to left from bottom to top, and end at index (2,2)
- Filter 3: start at index (2,2), move top to bottom from left to right, and end at index (last row - 1, last column - 1)

- Display the original data array and the three different filtered arrays on a 2x2 subplot grid with this layout:

Original	Filter 2
Filter 1	Filter 3

Instead of the `plot` command, display the images with the `heatmap` command. This command must be suppressed with a semicolon. For example: if you named your original pixel data as a variable `data`, your heatmap command to display the original data will be

```
heatmap(data);
```

Include a title for each heatmap, using only the filter name or "original data". Do not add any other plot formatting – use the defaults that appear with the `heatmap` command.

Hint: remember that `median` works differently on arrays than it does on vectors.

When your function is complete, run it to generate the final figure with the filtered images. When you are finished with your script, run it and save a PNG file of your figure window (see help if needed). Properly name the 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 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.
4. You only need to add teammates one time (regardless of the number of problems in the assignment or the number of resubmissions your team makes).

3. **Individually**:
   a. Complete your m-file and run it to get your results.
      - 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 and data file to the appropriate problem in the individual Gradescope assignment (see the submission list at the beginning of this problem).
      - Submit your deliverables once all your context problems are complete. Click here for help.
      - Do not submit any other files.

Grading

LOs: PC05, EPS02, MAT05, MAT08, MAT06

Team plan: 1 point

Individual assignment point value: 14 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	EPS02	MAT05	MAT08	MAT06	MAT06	MAT06
(1)	-100%	0.8	0.6	0.8	0.4	0.4	0.4
(2)	-25%	0.8	0.6	1.2	0.8	0.8	0.8
(3)	-10%	0	0	0.8	1	1	1
(4)	-15%	0	0	0	0.4	0.4	0.4
(5)	0	0	0	0	0.2	0.2	0.2
(6)	0	0	0	0	0	0	0
(7)	0	0	0	0	0	0	0
(8)	0	0	0	0	0	0	0

- Will grade the nested looping structure for each filter separately.

Grading Process