Concept Quiz 2

ENGR 132 – Summer 2021

Instructions – Read Thoroughly and Carefully

Academic Integrity

You must follow the course and Purdue's academic integrity expectations. You are allowed to use the following resources:

- MATLAB and MathWorks documentation
- Class material (homework, class slides, online videos, etc.)
- Internet searches

Do not share your instructions, answers, solutions, code, internet search links, or any other quiz material with any peer or electronic sharing site.

Saving Image Files

You will need to save figure windows as image files to submit to Gradescope. To save a MATLAB figure as an image file, click File > Save As from the figure window. **Always save the file as a PNG file**. Name the file using the format provided in the problem. The file name must always contain your Purdue career account login.

> **Hint**: Trouble viewing a full equation inside an instruction figure? Resize the figure window to see the full equation before you save the image. Three options to try: Drag the corner of the figure, use the quick edit tools in the figure, or use View > Property Editor. Video example: Resize a figure in MATLAB (Note: the process is the same for MATLAB Online or Desktop.)

Submission Requirements

Submit your solutions to **Concept Quiz 2 – ON TIME** on Gradescope. Your instructor will tell you the submission deadline for this quiz.

This assignment is timed. You have up to 1440 minutes (24 hours) to complete the quiz by the due date. Monitor the number of questions saved and the time remaining at the top of the quiz. **Save your answers after each problem**, or as many times as you want. Click **Submit & View Submission** at the bottom of the quiz to see what has been saved into Gradescope. Click **Resubmit** to add work or change your answers. You can re-submit your quiz as many times as you want until the due date. Your final submission will be graded. Click here and here for more help on the Gradescope submission interface.

As in your homework skills problems, you will submit a p-code function call with your PUID, your instruction text/information, and your solutions to the appropriate boxes or file submission areas. All instruction text will contain a ***run receipt*** with the format `su21_quiz2(PUID,problem_number) ran at DD-MM-YYYY HH:MM:SS`. Include this information with your instruction text.

Quiz Components

You will need the following items for this quiz. If you do not have access to these things, contact your instructor immediately.

- Your PUID number, MATLAB, access to Gradescope, internet access.
- This instructions document:
  - This problems document contains information you need for each problem on the quiz, but the problems are incomplete. To get the remaining information, you will use the file `su21_quiz2.p` in the same way you have been using the A0n_skills.p files in your homework.
- The problem generator code file `su21_quiz2.p`, which will display important information for each problem. Remember, you will not "open" the file. You will **run** it following these instructions.
  - Open MATLAB. Copy `su21_quiz2.p` into your current file path in MATLAB. To run the file, type the following command into the MATLAB Command Window prompt:

    ```
    >> su21_quiz2(PUID, problem_number)
    ```

    Where you replace PUID with your 8-digit Purdue University ID number (leave off the leading 00), and replace *`problem_number`* with the problem number indicated in the problems document. Using an incorrect PUID is considered Academic Dishonesty.
- A user-defined function file named `quiz2_matrixElems.p` that you will use in Problem 5.
- Template files for Problems 2, 3, 4, 5, and 7: Quiz2_Prob2_template.m, Quiz2_Prob3_template.m, Quiz2_Prob4_template.m; Quiz2_Prob5_template.m; and Quiz2_Prob7_template.m, respectively.

Tip for Success!

As soon as possible, run su21_quiz2.p with your PUID for all problems in this quiz. You want to make sure you get all your individualized instructions in place before you start working on the solutions to any problems.

The problem generator will give you information for each problem.

- All problems have instruction text and run receipts.
- Problem 3 has an instructional figure.
    - Save the figure window as a PNG. The filename format is Quiz2_Prob*X*_Instructions_*login*.png, where *X* is the problem number. See the problems below.
- In problem 2, make note of the CSV file that appears in your Current Folder.
- In problem 7, make note of the variables that are added into your workspace. You need to have the variables in your workspace for your solution to work. However, you *do not* submit the variables to Gradescope.

For each problem: Copy and submit all your instruction components to Gradescope in the appropriate problem areas.

What if something goes wrong and you get a syntax error when you run su21_quiz2.p for one of the problems? Take a screen shot of the function call and the error displayed in MATLAB. Send that to your instructor and GTA as soon as possible. You need to allow enough time for them to respond to you with a solution for the syntax issue. Continue working on other problems while you wait for a response.

Late submission

Late submissions will be allowed on this quiz. The late submission window gives you an additional 24 hours to complete any work you did not submit on time. **Each problem** submitted late will receive a 25% penalty. Immediately after the due date, **Concept Quiz 2 – LATE** will open in Gradescope with a new 24-hour window. Submit late work there. **Important!** You will not have access to your answers from the on-time submission. DO NOT DUPLICATE WORK IN THE LATE ASSIGNMENT ***unless*** you intend to replace the on-time work with the late submission. Any work submitted to the late assignment will overwrite your on-time submission and will be graded as late. See the figure below for a summary of the Gradescope submission windows.

In Gradescope	
Concept Quiz 2 – ON TIME	Open until the due date. Problems submitted here that are not duplicated in a late submission will be graded for full credit.
Concept Quiz 2 – LATE	Open after the due date, for 24 extra hours. Problems submitted here will be graded with a 25% penalty. Any problem submitted to both assignments will be graded here, with the penalty.

Quiz Problems

Problem 1 (6 points)

You will fix a while loop so that it executes the proper number of times. Run `su21_quiz2(`*`PUID`*`,1)` to get the while loop code. The code must be debugged. As written, the loop will run indefinitely (i.e., it is an infinite loop).

Part A:
Rewrite the while loop so that the loop will execute for the number of iterations specified in the instruction text. Do not change the initialization or calculation of `val`. You may hardcode values into the condition.

After rewriting the code, run it and determine the final value of `val`.

Part B:
Rewrite the code to be a `for` loop that achieves the same goal as Part A.

- Rewrite line(s) of code as necessary to change the loop type.
- Do not change the initialization or calculation of `val`.
- Remove redundant or unnecessary lines of code after changing the loop type.

Hint: You may find it helpful to create a script, but you will only submit code snippets to Gradescope.

Submit your instructions and solutions to Gradescope

- Enter your function call `su21_quiz2(`*`PUID`*`,1)` using your PUID in the appropriate box.
- Copy the instruction text from the Command Window and paste it into the appropriate box. **Include the run receipt.**
- Part A: Enter your code solution in the appropriate box.
- Part A: Enter your answer in the appropriate box.
- Part B: Enter your code solution in the appropriate box.

Problem 2 (14 points)

You work for an international company that produces procedure masks that are appropriate for use as personal protective equipment (PPE) as protection against the SARS Coronavirus 19 (COVID-19). You are an engineer at a US-based facility for this company that is considering selling the masks in Europe during the COVID-19 pandemic. To see if it is worth the cost, your task is to write MATLAB code that predicts the number of days required to achieve a target revenue, based on the predicted currency exchange rate between US Dollars ($) and Euros (€).

You have been given a script template named **Quiz2_Prob2_template.m**. Use the template to write a script to solve the problem. To solve the problem, you need more information that will be provided in the instruction text. Run `su21_quiz2(`*`PUID`*`,2)` to get the instruction text as well as a data file to use to solve the problem.

After running `su21_quiz2(`*`PUID`*`,2)`, you will see a new file in your current MATLAB folder named **Quiz2_currencyExchange_PUID.csv**, where PUID is the PUID number you entered into the p-code call.

Once your script is working, run it and then copy your results that are displayed to the Command Window and paste them into Gradescope.

Notes:

- Revenue is the money generated by the sale of the masks without considering the cost to make or ship the masks.
- Your code must contain a line to import the data from Quiz2_currencyExchange_PUID.csv.
- Assume that all masks produced in one day are sold that day at the indicated exchange rate.
- Follow good programming standards.

Submit your instructions and solutions to Gradescope

- Enter your function call `su21_quiz2(`*`PUID`*`,2)` using your PUID in the appropriate box.
- Submit your data file (Quiz2_currencyExchange_*PUID*.csv).
- Copy the instruction text from the Command Window and paste it into the appropriate box. **Include the run receipt.**
- Enter your script's results in the appropriate box.
- Submit your script m-file with an appropriate name (Quiz2_Prob2_*login*.m).

Problem 3 (8 points)

You will be given a flowchart and must translate that flowchart into a function. Run `su21_quiz2(`*`PUID`*`,3)` to see the flowchart in a figure window. The flowchart contains some shaded shapes that have variables that are defined as numbers in the Command Window instruction text. Hardcode those numbers in your translation code (i.e., do not use the variable letters A, B, etc.).

- 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/or instructor for help.
- Expand the figure window as needed to better read the flowchart.
- Translate the flowchart exactly as presented, even if you see another way to achieve the same outcome.

Note that the flowchart represents a user-defined function that requires a function definition line with one input argument and no output argument. Name the function using the function name stated in the instruction text. Assume x is a scalar. Complete your flowchart translation in the provided template: **Quiz2_Prob3_template.m**. Programming standards will not be assessed.

Save an image of your flowchart
Save an image of your flowchart figure as a PNG for submission. Name the image file using the format: Quiz2_Prob3_Instructions_*login*.png.

Submit your instructions and solutions to Gradescope

- Enter your function call `su21_quiz2(`*`PUID`*`,3)` using your PUID in the appropriate box.
- Submit the image file with the flowcharts (Quiz2_Prob3_Instructions_*login*.png).
- Copy the instruction text from the Command Window and paste it into the appropriate box. **Include the run receipt.**
- Submit the m-file with your flowchart translation with the appropriate name.

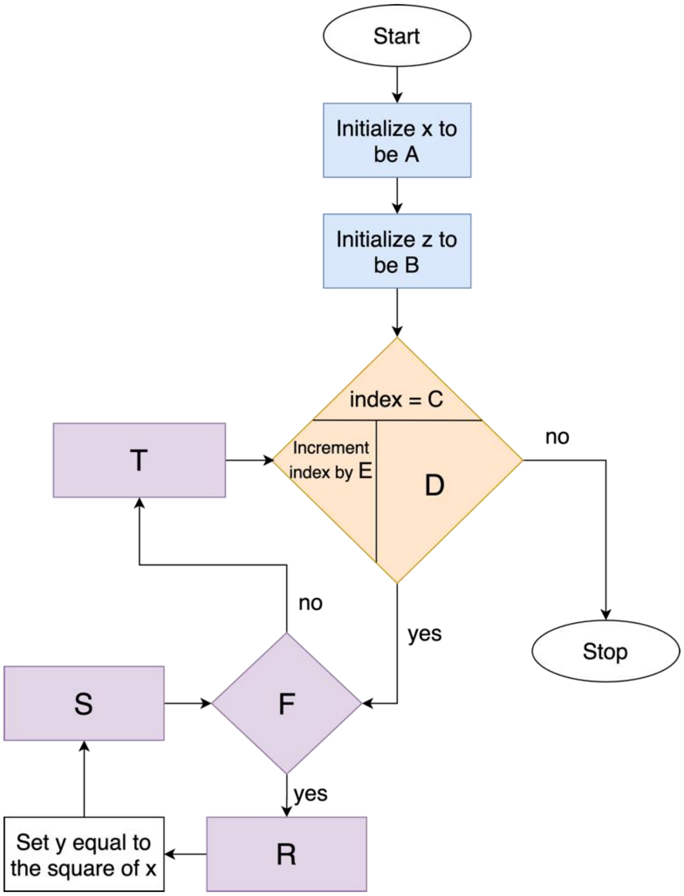
Problem 4 (14 points)

Consider the flowchart shown.

Notice that the shaded shapes in this flowchart have missing information (either missing numerical values, or missing instructions). Run `su21_quiz2(PUID,4)` to get the instructions that go into each missing shape. Use that information as you code the flowchart. The flowchart contains some shaded shapes that have variables that are defined as numbers in the Command Window instruction text. Hardcode those numbers in your translation code (i.e., do not use the variable letters A, B, etc.).

You have been given a script template named **Quiz2_Prob4_template.m**. Use the template to write a script to solve the problem.

When your script is complete, run it to find the final values of the variables x, y, and z. Fill in your answers and information on Gradescope.



Submit your instructions and solutions to Gradescope

- Enter your function call `su21_quiz2(`*`PUID`*`,4)` using your PUID in the appropriate box.
- Copy the instruction text from the Command Window and paste it into the appropriate box. **Include the run receipt.**
- Enter your final value for **x** in the appropriate box.
- Enter your final value for **y** in the appropriate box.
- Enter your final value for **z** in the appropriate box.
- Submit the m-file with your flowchart translation (Quiz2_Prob4_*login*.m).

Problem 5 (12 points)

You must write a nested looping structure (i.e., a loop nested within a loop) that will create a matrix named `mat1` that contains copied elements from a larger matrix. The larger matrix is a 20x20 matrix that is contained within the provided p-code `quiz2_matrixElems.p`. You had a chance to learn about this function in class. Refer to those instructions for more details, if necessary.

To return a single element at a given array index, use this function call with syntax that uses two integer inputs and one output:

```
>> val = quiz2_matrixElems(row_index, column_index)
```

You can only retrieve ONE ELEMENT AT A TIME from this function. It will not accept a range of indices.

You will need a starting array index and an ending array index to identify which values will be added into `mat1`. Run `su21_quiz2(`*`PUID`*`,5)` to get the array index for the starting element and the array index for the ending element and the dimensions of your new matrix, `mat1`. Figure 1 has examples of the four possible ways in which you will copy the elements from a subset of `quiz2_matrixElems` to `mat1`. For `mat1`, the direction of movement for placing values is ALWAYS left to right, then top to bottom.

Code your solution using the provided script template **Quiz2_Prob5_template.m**. You may hardcode numeric values given in the instruction text. Your looping structure only needs to work for the indices and direction you were given in the instruction text. Make the last line of your code be `disp(mat1)` to display the final matrix result. Programming standard will not be assessed.

Run your final script to get the display for mat1. Copy and paste that result for mat1 into the appropriate box in Gradescope.

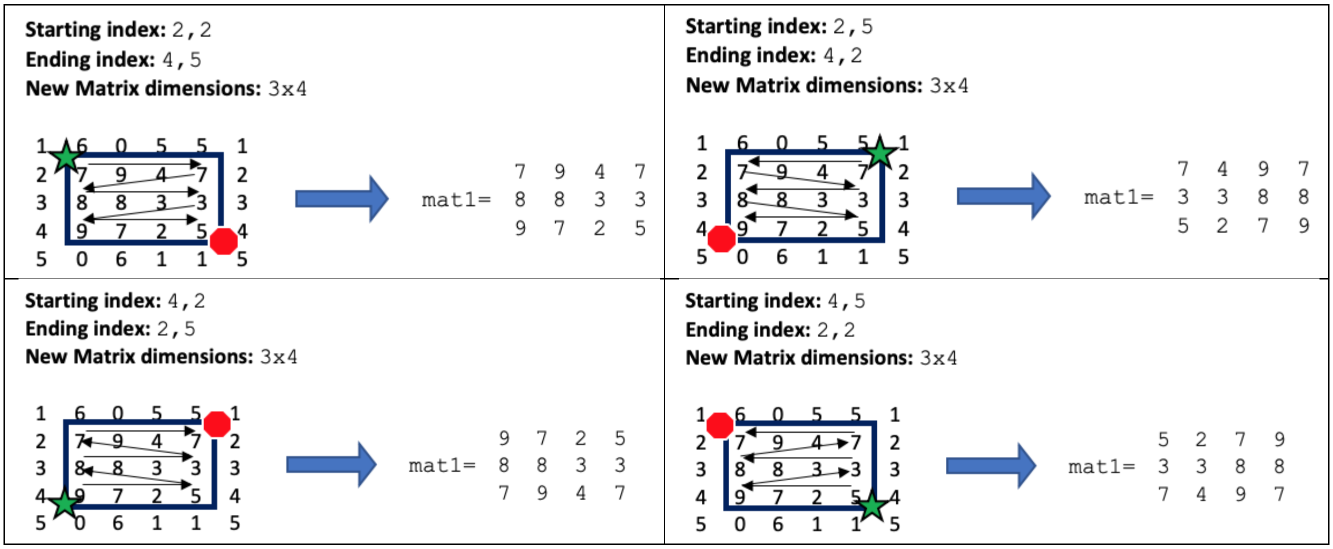


Figure 1. Four examples of taking elements from larger matrix to fill `mat1`. The direction you will pull elements from `quiz2_matrixElems` is dependent on your starting and ending indices. The direction of movement for placing values in `mat1` is ALWAYS left to right, then top to bottom.

Submit your instructions and solutions to Gradescope

- Enter your function call `su21_quiz2(`*`PUID`*`,5)` using your PUID in the appropriate box.
- Copy the instruction text from the Command Window and paste it into the appropriate box. **Include the run receipt.**
- Enter your result for `mat1` in the appropriate box.
- Submit the m-file with your looping structure (Quiz2_Prob5_*login*.m).

Problem 6 (6 points)

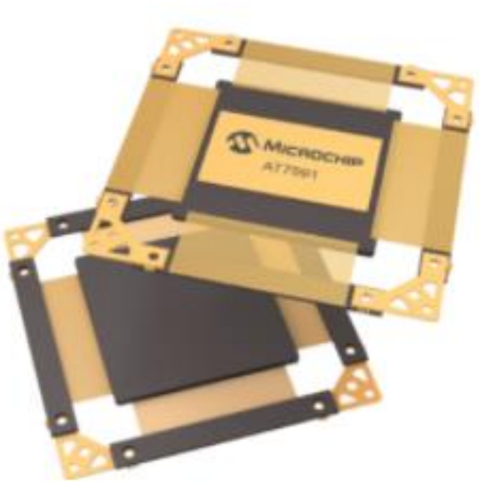
You must convert a given script to a user-defined function. Run `su21_quiz2(PUID,6)` to display the script to the Command Window. The function output arguments are all the variables assigned in the `%% OUTPUT CALCULATIONS` section of the script. You will need to determine the number of input arguments for the function by examining the script.

Make <u>no changes</u> to the script code. You only need to write a function definition line. Name the function using the name provided in the instruction text.

Submit your instructions and solutions to Gradescope

- Enter your function call `su21_quiz2(PUID,6)` using your PUID in the appropriate box.
- Copy the instruction text from the Command Window and paste it into the appropriate box. **Include the run receipt.**
- Enter your function definition line in the appropriate box. Do not include any other lines of code.

Problem 7 (15 points)



Space radiation can cause malfunctions in microchips which can be really problematic for spacecraft. These malfunctions happen in two ways: repeated exposure over mission lifetime and single high-intensity events. Microchips are rated to allow a certain amount of repeated radiation exposure, known as total ionizing dose (TID). Random radiation events, such as solar flares or cosmic rays, can allow radiation to move through the microchip and cause a single event latch-up (SEL). Microchips are shielded from these high-intensity radiation events. A microchip's linear energy transfer (LET) threshold defines the level of shielding, with higher LET values providing more protection.

You are an engineer responsible for finding appropriate microchips for space missions. You must write a user-defined function that will identify a list of possible microchips from a catalog selection based on two mission criteria: TID threshold and LET threshold under which there are no single event latch ups at 125 degrees Celsius.

Run `su21_quiz2(PUID,7)` to obtain a catalog of microchips and their specifications. You will see 3 vector variables appear in your workspace. The variables have the same dimensions.

`catalog_chipname`	Nx1 string vector where each element is the name of a microchip
`chip_LET`	Nx1 number vector where each element is the microchip's LET threshold (MeV.cm^2/mg)
`chip_TID`	Nx1 number vector where each element is the microchip's TID threshold (krad)

Use these variables as you write your function. You will also see test case values. You will use those after you write your function.

You function must do the following. Use the included template **Quiz2_Prob7_template.m**.

- Name your function `Quiz2_Prob7_login` where *`login`* is your career account login.
- Accept five (5) input arguments and return zero (0) output arguments.
  - Input 1: the minimum LET threshold the microchip must survive (in MeV.cm^2/mg)
  - Input 2: the minimum TID the microchip must survive (in krad)
  - Input 3: `catalog_chipname`
  - Input 4: `chip_LET`
  - Input 5: `chip_TID`

- Confirm the input arguments are valid.
  - Ensure the input LET greater than 0 and does not exceed the maximum LET threshold in the microchip catalog. Display a meaningful message if it is outside this range.
  - Ensure the input TID is greater than 0 and does not exceed the maximum TID in the microchip catalog. Display a meaningful message if it is outside this range.
- Determine which microchips meet the input TID and LET values.
  - **Hint**: String vectors work like numeric vectors. You can use built-in functions like `numel`, and you can find elements using indexing.
- Display the results to the Command Window.
  - If there are chips that meet the TID and LET criteria, list them by name in the command window (use the `disp` command to display the string array of final chips)
  - If no chips meet the criteria, then display the statement "No chips meet the input criteria."
- Follow good programming standards.
  - Use the provided template. Save with the appropriate file name.
  - You may hardcode 0 for the input validity check. Do not hardcode any other numbers in your function.

Run your function with the test case provided. Copy the **function call AND the printed results** and paste them into Gradescope.

- **Hint**: If you need to re-add `catalog_chipname`, `chip_LET`, and `chip_TID` to your Workspace, rerun `su21_quiz2(PUID,7).`

Submit your instructions and solutions to Gradescope

- Enter your function call `su21_quiz2(`*`PUID`*`,7)` using your PUID in the appropriate box.
- Copy the instruction text from the Command Window and paste it into the appropriate box. **Include the run receipt.**
- Enter your results in the appropriate box.
- Submit your function m-file with an appropriate name (Quiz2_Prob7_*login*.m).