ENGG*1420 Lab 0: Pointers Refresher

For this lab, you can use C programming language.

At the end of the lab, you will be required to upload your file(s) on Dropbox and then show your implementation to the TA to get your grade.

This is an open-book lab. You can use the resources provided to you in the course (videos/notes/lectures) as well as Google to help with this lab.

Part 0:
For this first part, you are required to create a small program using the following instructions:

1. You were provided a lab0_skeleton.c make sure you download it.
2. Open lab0_skeleton.c
3. Follow the instructions in the skeleton to populate main() to create a control switch function that displays this list of options available to the user:

```
Please Select a number from the following options:
1. Q1
2. Q2
3. Q3
4. Exit
```

   asks the user for input as an int (question number), checks the value provided, and depending on what the user gave calls the appropriate function (for now the functions can just print their name until we populate them in later parts). If the user enters an invalid value, return an error and reprompt them. Repeat this until the user enters 4.

4. Compile and run.

Part 1:
Follow the instructions in part one in the skeleton to populate partOne(int a). This may require you to modify the main in order to pass parameters/print results.

Part 2:
In partTwo() create a function that will ask the user for an array size, use malloc to allocate memory for an array of said side.

1. Must check if memory was successfully allocated
2. If it was not, the program should return an error and exit gracefully
3. If memory is successfully allocated, populate an array with random numbers (using a random number generator)
4. Display array content

Part 3 :

In partThree() similar to partTwo create a function that will ask the user for a number or rows and several columns, and use malloc to allocate memory for a 2D array of said side.

1. Must check if memory was successfully allocated.
2. If it was not, the program should return an error and exit gracefully
3. If memory is successfully allocated, populate an array with random numbers (using a random number generator)
4. Display array content

For your Demo:

you have to show the

1. Commands in steps 1-3
2. Code
    a. Code must be commented on and styled as the skeleton provided
    b. The code must be readable
    c. You must walk the TA through your code and explain what you did
3. Compile and Run the code and show the output to the TA.