Practice Exam based on Spring 2020 Exam 2 (Moving Window)

Write a program that will create a random matrix, extract a column, then follow steps to modify the values in the vector.

Function

Create a function named **ChangeColumn** that looks through a column of a matrix and creates a final matrix that stores the positions of the changed values (index) in column 1, and the modified values in column 2.

First pull out the specific column and store as **NEWCOL**. Then do the following two checks:

- If the first value in **NEWCOL** is not smallest value, replace it with the minimum value of **NEWCOL.**
- If the last value in **NEWCOL** is not the largest value, replace it with the maximum value of **NEWCOL.**

For the remaining values, do the following check:

- If the current value is NOT greater than or equal to the previous value (row above) or NOT less than or equal to the next value (row below), replace the current value with the average of value located one row above and one row below. Otherwise, leave the current value unchanged.

NOTE: The number of rows in output is equals the number values replaced.

Function Inputs:

1. Start matrix
2. Column number to store and change

Function Outputs

1. Final matrix storing changes

The function header should be formatted similarly to the following:

```
function [out1] = ChangeColumn(in1,in2)
```

Remember you are free to use whatever variable names you want, but they must be listed in the same order as given in the input/output lists provided above.

Main Script

Data Entry and Validation:

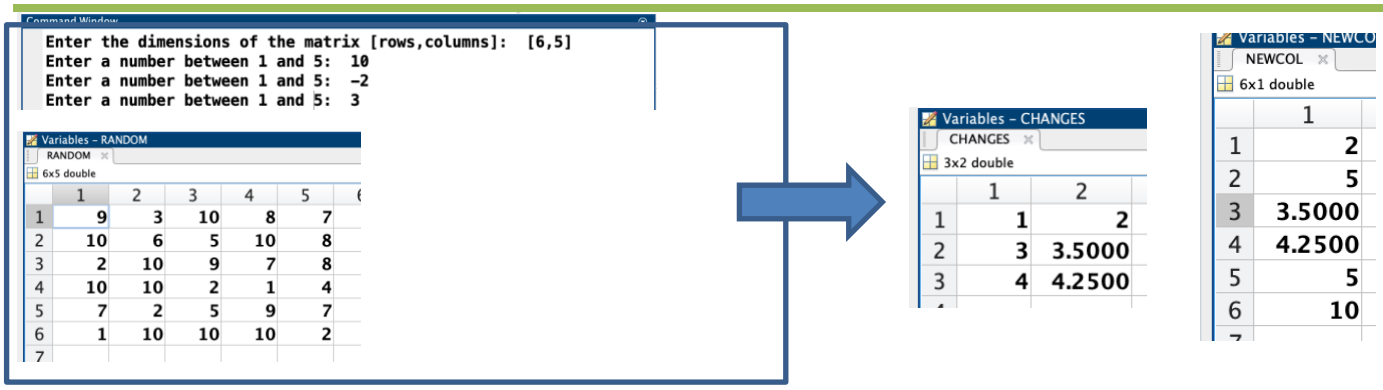
- Ask the user to enter the dimensions of the random matrix to be generated, as a 1 x 2 vector [rows, columns] and store it in **DIM**.
- Create a random matrix of integers between 1 and 10 using the dimensions provided by the user.
- Ask the user to enter a number between 1 and the number of columns and store this value in a variable named **COLUMN**

If the user enters a **COLUMN** that is outside the number of columns specified (1 to **COLUMN**), ask the user to enter another value. Keep asking the user. If after the 4 attempts (counting the first attempt), the user has not entered an appropriate value, give a warning and set **COLUMN** equal to 1.

Apply Function

Using the function, **ChangeColumn**, determine the change matrix and save the results a .mat filed named **Exam2Changes.mat**.

Sample Output:



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

	1	2
1	1	2
2	3	3.5000
3	4	4.2500

	1
1	2
2	5
3	3.5000
4	4.2500
5	5
6	10