Drill Problem #1

Function Name: meanNonZero

Inputs:

1. (*double*) A 1xN vector of numbers

Outputs:

1. (*double*) The average value of all nonzero elements of the input vector

Function Description:

Imagine you are an engineer performing an experiment for research, and you have some super fancy equipment recording numerical data values at a rate of 1000 values per second. You look at the data after one minute, and for some reason there are a lot of 0's recorded (due to sensor malfunction). You need to find the average of all the data points, but all those random 0's are make the average artificially low. So you write a MATLAB function…

Write a MATLAB function to find the mean value of all nonzero elements in any given vector. The vector can be of any length. There can be any number of 0's in the vector.

Drill Problem #2

Function Name: checkPalindrome

Inputs:

1. (*string*) An input string that may or may not be a palindrome

Outputs:

1. (*logical*) A logical indicating whether or not the input string is a palindrome

Function Description:

You recently read the sentence “A man, a plan, a canal Panama,” and you realized that if you read it backwards, it still reads “A man, a plan, a canal, Panama” (excluding the location of punctuation), meaning this sentence is an example of a palindrome. You decide to make a MATLAB function that will check an input phrase to see whether it is a palindrome. The input to your function is guaranteed to have only letters and spaces (no punctuation); however, spaces and capital letters will not affect whether or not something is a palindrome. In other words, both “racecar” and “Race car” should output `true`, indicating that they are both palindromes.

Hints:

- The `strcmpi()` function may be helpful.

Drill Problem #3

Function Name: checkPassword

Inputs:

1. (*char*) A string containing a user-provided password

Outputs:

1. (*logical*) A logical specifying if the password provided meets the given criteria

Function Description:

In the recent security leaks of various corporations, passwords have to be more complex than ever (and since we're at Tech, we all know this). Write the MATLAB function checkPassword that takes in a string containing a password and determines if it meets the criteria. It should output `true` if it does meet the criteria and `false` if it does not.

Criteria:

- The string must contain at least 8 characters (8 OR above).
- The string must contain at least 1 number, 1 lowercase letter, and 1 uppercase letter.
- The string must contain at least 1 symbol that ranges from a space to a forward slash (/) on the ascii table.

Further, if the password to be checked contains the word password (or any of its upper/lowercase permutations), it automatically does not pass the check and should return false. Example:

- passworD1$ – false
- 20/passWord15 – false
- PaSSword!3@ - false
- p%as2015Sword – true

Notes:

- It is acceptable to use symbols that are not in the specified range, but there must be at least 1 symbol that is from the specified range.

Hints:

- The `strfind()` function may be helpful.

Drill Problem #4

Function Name: camelCase

Inputs:

1. (*string*) A phrase to be converted to camel case

Outputs:

1. (*string*) The phrase converted to camel case

Function Description:

Write a MATLAB function that will take a phrase and convert it to camel case. Specifically, this function should make the first letter of every word capitalized, except for the very first letter, and all other letters lowercase. The function should also remove all the spaces from the input phrase. So, the string `'MATLAB is awesome'` should become `'matlabIsAwesome'`.

Notes:

- This problem must be done using only logical indexing.

Drill Problem #5

Function Name: theRatio

Inputs:

1. (*double*) A 1XN vector of the number of female students enrolled at Georgia Tech
2. (*double*) A 1XN vector of the total number of students enrolled at Georgia Tech
3. (*double*) A 1XN vector corresponding to the year of enrollment

Outputs:

1. (*double*) A 1X2 vector of the years corresponding to the largest growth in the ratio

Function Description:

We have all experienced, at one time or another, the rather prevalent "ratio" of females on Georgia Tech's campus. Ever since the first woman stepped foot on Tech's campus, Tech has boasted about its tremendous strides in increasing the number of women in the field of engineering. Being the MATLAB wiz that you are, you have decided to calculate Tech's achievements in female recruitment and prove in which year GT was most successful.

Given a vector of the number of women enrolled at Tech per year, a vector of the total number of students enrolled at Tech per year, and a vector corresponding to the year of enrollment, write a function called `theRatio` that will output a 1X2 vector of the two years between which there was the largest increase in the ratio of women to total students at Georgia Tech.

Example:

Say in 1990 there were 580 females at GT and 1673 total students. In 1991 there were 602 females and 1721 total students. The change in the ratio between these two years is calculated by dividing the number of females by the total number of students for each year, and finding the difference between these two values:

```
ratio = (602/1721) – (580/1673)
         ratio = 0.0031
```

If this were the largest positive change, then the output year vector would look like

```
[1990, 1991].
```

Notes:

- It is guaranteed that the ratio growth will not be the same between multiple years
- The years will always be listed in chronological order

Drill Problem #6

Function Name: criminalMinds

Inputs:

1. (*logical*) Vector of suspect #1’s answers to a lie detector
2. (*logical*) Vector of suspect #2’s answers to a lie detector
3. (*logical*) Vector of suspect #3’s answers to a lie detector
4. (*logical*) Vector of suspect #4’s answers to a lie detector

Outputs:

1. (*char*) Sentence stating which suspect is lying

Function Description:

After all of those years of reading Nancy Drew and watching Bones, you realize that your true passion lies in justice and bringing criminals to light. After tedious years of training with the FBI, you are finally working a case, and you have four suspects—only one of which is the true criminal. You give them each a separate lie detector test and plan to use the results to find which of the four suspects is lying to you. Each suspect who is telling the truth will give the same corresponding yes or no (`true` or `false`) answers, since that is how real life innocent suspects behave, while the suspect who is lying will not corroborate his/her answers with the other three. Since you were a pro at MATLAB back in your engineering days, you decide to write code to assist you in finding the criminal.

Write a function with four inputs, where each input is a logical vector corresponding to the answers a suspect gives on the lie detector. Three of the suspects will have the exact same answers, but one suspect’s answers will be slightly—or completely—different than the others’ answers. Using your knowledge of logical indexing and masking, find which of the four suspects is lying and, thus, is the criminal. The output string will be of the form `'Suspect #_ is lying`.' Where the ‘_’ corresponds to suspect number who is lying and the number is from the input order.

Notes:

- The suspect who is the liar will have at least one answer that is different from the other suspects’ answers, but could differ up to every answer.