Drill Problem #1

Function Name: varInfo

Inputs:

1. (*double*) A number or vector of any length.
   OR
   (*char*) A string of any length.

Outputs:

1. (*char*) An output string describing the input variable.

Function Description:

Write a MATLAB function to identify the class of the input variable and to display its value. Your function should output a string of the form:

```
This variable is of class ____ and has a value of '____'.
```

For example, if the input variable is the number 25, then your function should output the string: `This variable is of class double and has a value of '25'.` If instead the input variable is the string `'CS 1371'`, then your function should output the string: `This variable is of class char and has a value of 'CS 1371'.`

Notes:

- Make sure that your output variable matches the answers to the solution file EXACTLY. Any extra/missing spaces or characters will result in a 0 for this problem.
- Using the MATLAB function `num2str()` on a string will not affect its data value and will output the same string. You will want to use `num2str()` on the vector to print the vector to a string with spaces between the indices. Do NOT use `mat2str()` for this function.
- The period at the end of the output strings shown above is part of the string.

Hints:

- **Use `isequal()` to know if your output and the solution file output are exactly the same.**
- The `class()`, `num2str()`, and `sprintf()` functions will be useful.
- To get an apostrophe into a string, type two apostrophes into MATLAB.

Drill Problem #2

Function Name: multiVar

Inputs:

1. (*double*) A vector of x values
2. (*double*) A vector of y values

Outputs:

1. (*double*) A vector of the function $z(x, y)$ evaluated at corresponding x and y values.

Function Description:

Write a MATLAB function that will evaluate the shown multi-variable function for *vectors* of x and y values. Your function should output a *vector* of the function evaluated for corresponding x and y values. For example, the third index of your output vector should be the following function evaluated using values from the third index of the input x and y vectors.

$$z(x, y) = \frac{xy - \frac{\sqrt{x}}{y}}{3x + y}$$

Notes:

- Recall that the use of "dot operators" is important for mathematical operations with vectors, excluding addition and subtraction.

Drill Problem #3

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 whiz 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 consecutive 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.
- You do not have to account for any disparate spacing between year intervals.

Drill Problem #4

Function Name: interweave

Inputs:

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

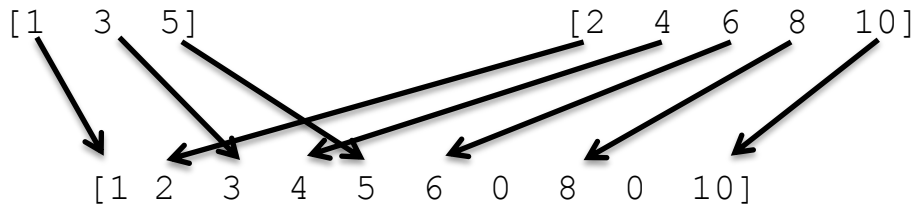
Outputs:

1. (*double*) A vector of the two input vectors mixed together.

Function Description:

Write a function called `interweave` that takes in two vectors and then outputs a larger vector where the odd elements contain the values from the first vector, and the even elements contain the values from the second vector. If one vector is longer than the other, then the odd or even indices that wouldn't otherwise have values should be filled with zeros.

For example, given the following two vectors for inputs:



The above vector would result from interweaving. Note the two zeros at the end of the vector, since the first vector was not as long as the second vector.

Hints:

- The length of the output vector should be twice the length of the longest input vector.
- The `max()` and `zeros()` functions may be useful.

Drill Problem #5

Function Name: gradeSorter

Inputs:

1. (*double*) A vector of student grades.
2. (*char*) A string of first initials corresponding to the grades.
3. (*char*) A string of last initials corresponding to the grades.

Outputs:

1. (*char*) A string of first and last initials sorted by corresponding grade.

Function Description:

You try to get on your professor's good side by mentioning your awesome MATLAB skills. Unfortunately, it backfires because he then ignores FERPA and asks you to write code to sort the first and last initials of the students in the class according to their grades.

Your professor gives you a vector of the student grades, a string of first initials corresponding to the grades, and a string of last initials corresponding to the grades. Given these inputs, write a function that outputs a string with the combined first and last initials of every student, with spaces in between, that is sorted by the students' grades in descending order. To clarify, an output string might look like this: `'MS MO ND AS'`, where the student `MS` had the highest grade, and the student `AS` had the lowest. Notice that there are spaces in between the initials of each student but not after the last student, and note that there is no punctuation in the string.

Notes:

- The `sort()` function will take care of sorting equivalent grades.
- You will not have a tie in the grades.
- The ASCII value of a space is 32. Any other values that appear as a space will be marked as incorrect and result in zero credit.

Hints:

- The second output of `sort()` will be useful.

Drill Problem #6

Function Name: caesarSalad

Inputs:

1. (*char*) A 1xN string of a single word.
2. (*double*) An integer describing the shift, or the “shift number”.

Outputs:

1. (*char*) The input word encoded using the Caesar cipher.

Caesar Cipher Information:

The Caesar cipher is named after Julius Caesar, who, according to Suetonius, used it with a shift of three to protect messages of military significance. It is unknown how effective the Caesar cipher was at the time, but it is likely to have been reasonably secure because most of Caesar's enemies would have been illiterate and others would have assumed that the messages were written in an unknown foreign language.

Caesar ciphers can still be found today in children's toys such as secret decoder rings. A Caesar shift of thirteen is also performed in the ROT13 algorithm, a simple method often used to obscure text such as joke punchlines and spoilers online.

In the Caesar cipher, each letter is shifted by the specified amount. For example, if the shift is 3, then the letter `'a'` would be coded as the letter `'d'`.

Function Description:

Write a function that takes in a string of a single word and uses the Caesar cipher with the input shift number to encode it. Only lower case letters will be included in the input string; any other characters—such as spaces, periods, commas, etc.—will not be included as part of the input.

Notes:

- The Caesar cipher should work for both positive and negative integers that indicate the shift given by the second input.
- There is no limit to the value of the shift number in the second input.

Hints:

- The `mod()` function will be useful.