

Function Name: goLong

Inputs:

1. (*double*) A 1x4 vector representing coordinates (x1, y1, x2, y2)

Outputs:

1. (*char*) A string stating the distance between (x1, y1) and (x2, y2)

Background:

It's the first football game of the year and you're sitting in the bleachers, but instead of watching the game, you're using the MATLAB mobile app. When you finally look up from your phone, you see WR Ricky Jeune streaking down the field wide open! But does new QB Matthew Jordan have the arm strength to pass that far?

Function Description:

You have been given the (x,y) coordinates of the two players on the football field, represented by a vector, where each coordinate is listed in the order [x1, y1, x2, y2]. Use your newly acquired MATLAB skillz to concisely state the distance between the two players as a string! Your output string should be in the following format:

'(<x1>, <y1>) is <distance> units from (<x2>, <y2>).'

In case you forgot, the Euclidean distance formula between two points is:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Example:

```
[out] = goLong([7 8 6 5])
```

```
out =
```

```
'(7.00, 8.00) is 3.16 units from (6.00, 5.00).'
```

Notes:

- In the output string, each number should be displayed to two decimal places.
- Use %.2f to display two decimal places in a formatted string.

Hints:

- The sprintf() function will be useful here.

Function Name: buzzCase

Inputs:

1. (*char*) A 1xN string to be converted

Outputs:

1. (*char*) The input string in 'Buzz case'

Background:

It's football season! With our opening game this week, we need your help! We are going to take a string and add a little GT flair to it by converting it to Buzz case!

Function Description:

Write a function in MATLAB that takes in an input string and converts it to Buzz case. What is Buzz case you ask? Well, to convert a string to Buzz case, there are a couple of steps you need to take:

- 1) The entire string should be made lower case.
- 2) All spaces should be replaced with a series of Z's.

The number of Z's that replace each space should be equal to the total number of G's and T's (either upper or lowercase) in the original string. The final string should have no spaces and be all lowercase except for the capital Z's where the spaces used to be. Soon the string will be buzzing with GT spirit!

Example:

If the input string is 'Go Jackets', then output string should be 'goZZjackets'. The output string has no spaces and is all lower case except for the Z's. There are two Z's because there are two total G's and T's in the original string.

Notes:

- There will not be any punctuation in the input string.
- If there are no G's or T's in the input string, the output string should look like the input string lowercased with no spaces.

Hints:

- The `strfind()` and `strcmp()` functions may be useful.

Function Name: `rebelCipher`

Inputs:

1. (*char*) A string to be encoded
2. (*double*) The encryption key

Outputs:

1. (*char*) An encoded string

Background:

Help us CS 1371 Student! You are our only hope! The Rebel Alliance has discovered the Galactic Empire plans for building a massive super weapon that they can use to usher in a new era of tyranny. (OH NO!) An Imperial pilot has information on the location of these plans, but no way to get it into Rebel hands without the Empire finding out. Luckily, using your superb MATLAB skills, you can write a function to encode the pilot's message, thus saving the galaxy!

Function Description:

The encoder (or cipher) you will be creating will be based on a shift cipher. A shift cipher is where each letter of the alphabet is shifted over a certain number of places based on the encryption key. For example, if the encryption key was 3, then 'a' would become 'd'. The shift also wraps around the end of the alphabet, so the letter 'z' would shift to become the letter 'c'.

Your function takes in a string to be encoded and an encryption key representing how far each letter will be shifted for your cipher. It then performs the following actions in order on the input string:

1. Convert the entire string to upper case, since secret messages only have one case.
2. Perform the shift based on the encryption key.
3. In some cases, the cipher will shift backwards through the alphabet and/or cycle through multiple times. Therefore, you should also output the equivalent positive shift between the values 0 and 25, and add that number to the beginning of the encoded message.

Example:

If given the string `str = 'abcDEF'`, and encryption key = 29, your function should output the string `msg = '3DEFGHI'`.

Notes:

- The function should work for both positive and negative integer shifts of any magnitude.
- The input string is guaranteed to consist of only letters (no punctuation or spaces).

Hints:

- The `mod()` and `num2str()` functions will be useful.
- For Fun: Try some classic Tech phrases and the CS 1371 TA test team motto in the solution function!

Function Name: braid

Inputs:

1. (*char*) A 1xN string
2. (*char*) A 1xN string
3. (*char*) A 1xN string

Outputs:

1. (*char*) A 1x(3*N) string combining the input strings

Function Description:

Oh no! The radios used to call plays to the quarterback have mysteriously stopped working at Mercedes-Benz stadium! Luckily, Coach Paul Johnson always has a plan*. He and two assistant coaches encode the plays into three separate message boards for everyone to see. QB Matthew Jordan quickly catches on, but he didn't learn MATLAB as part of his BBA, so he has DB Zach Matthews, an ME, write a decoder for him. Here are the rules for decoding.

Your three input string components will be the same length and should be “woven” or “braided” together so that the first string will be at the 1st, 4th, 7th... indices and the 2nd string will be at the 2nd, 5th, 8th... indices and the 3rd string will be at the 3rd, 6th, 9th... indices. Once you have created this string, reverse it to find the hidden message.

Example:

Given the three strings 'GR PXE', 'NTEMEH' and 'ISLA T', the string created by braiding the strings together will be 'GNIRTS ELPMAXE EHT'. Continuing with this example, your final output string would read:

'THE EXAMPLE STRING'

Hints:

- Think about using the colon operator to create an index vector.
- For even more funzies, you could write the inverse of this function (a function that takes a string and encodes it). Then you can be just like your icon, Coach Paul Johnson!
- *Plan developed for use against u(sic)ga, as they would never catch on.

Function Name: compAIRContour

Inputs:

1. (*double*) A 1xN vector representing the air pressures in the current stadium
2. (*double*) Another 1xN vector representing the air pressures of the GT practice field

Outputs:

1. (*logical*) Whether or not the "contour" of the two vectors is the same

Background:

With four seconds on the clock, a tie game, and the ball at the 40 yard line, most coaches would take a knee and play for overtime, but not the great Coach Paul Johnson! CPJ trots out freshman PK Brenton King to attempt the 57 yarder, eclipsing the 53 yard mark set by Harrison Butker. As a rookie, King does not have experience to rely on like Butker did. Luckily, he has something better: an intimate knowledge of the air pressures in the stadium. Now, he needs to figure out whether the wind contour here matches the contour back at the GT practice facility, so he can nail this kick and become the stuff of legends.

Function Description:

For the sake of this problem, we will define the *contour* of a vector to be whether adjacent elements of the vector are increasing or decreasing. Therefore, any vector defines a sequence of up and down contours based on the difference between adjacent values. The contour of two vectors is the same if they both have the same pattern of increasing and decreasing elements. For the contour to be the same, the vectors DO NOT have to have the same values, nor does the amount by which elements increase or decrease have to be the same; only the pattern of increasing and decreasing matters.

Consider the following vectors

$$v1 = [2, 4, 3]; v2 = [-2, 10, 8]$$

These two vectors have the same contour because they both follow the pattern [increasing, decreasing]. Now consider these two vectors:

$$v3 = [5, 3, 1]; v4 = [1, 4, 1]$$

These vectors do not have the same contour. The first one follows the pattern [decreasing, decreasing] while the second one follows the pattern [increasing, decreasing]. The function should output true if and only if the two vectors have the same "contour".

Notes:

- Both vectors will always be the same length.
- A single number has the same contour as any other single number.
- **Make sure that your output is class logical and not double. You will get the problem wrong if you output a double 0 instead of a logical false.**

Hints:

- The `diff()`, `abs()`, and `isequaln()` functions will be useful.
- If N is a number, the sign of N can be found using the formula: $\text{sign } N = \frac{N}{|N|}$