

Notice

Please pay attention to the class of the necessary outputs on the homework and to the class of the variables in the ABCs. Most of this week's homework problems require a char output. This means a vector of type 'char', NOT a string array. Additionally, one problem requires a logical (true or false) output. Be sure that your output is of type logical, not the double 1 or 0. If you are unsure, you can always check the class of your variable using the `class()` function. As a general rule for CS1371, use vectors of type 'char', and not MATLAB's string array class.

Happy Coding!
~Homework Team

Function Name: weatherStats**Inputs:**

1. (*double*) A 1xM vector of temperature values throughout the day
2. (*char*) A character vector describing the day's weather

Outputs:

1. (*char*) A formatted character vector describing the day's weather statistics

Function Description:

For your new job at NOAA, you need to process weather data. You are given a vector of temperatures recorded during the day, which you must analyze to determine the maximum and average temperature for the day. You also have a written description of the day's weather from NOAA, which is written in all caps. This description needs to be turned to lowercase so it is easier to read. Finally, combine all of this information into a formatted character vector.

The outputted vector should be in the following format:

```
'Today's weather was <description>, with a high temperature of <max temperature> degrees and an average temperature of <avg temperature> degrees.'
```

Example:

```
[out] = weatherStats([23, 26, 28, 35, 30], 'FROSTY')
out => 'Today's weather was frosty, with a high temperature of 35
degrees and an average temperature of 28 degrees.'
```

Notes:

- The vector of temperatures can be any length.
- Average and max temperatures should be rounded to the nearest whole number.
- The weather description may contain spaces.
- You can use two single apostrophes to get a single apostrophe in a character vector.
 - `var = 'Today''s weather'`
`var => Today's weather`

Hints:

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

Function Name: sweaterWeather

Inputs:

1. (*char*) A 1xN vector of characters
2. (*char*) A second 1xN vector of characters

Outputs:

1. (*char*) A new 1x(2*N) woven vector

Background:

Upon discovering the forecast for chilly weather this morning, you don your favorite striped sweater! You notice the pattern of alternating threads woven into your sweater, and (being a MATLAB enthusiast) you decide to write a function that can weave vectors together in a pattern much like that of the threads in your own dashing cardigan.

Function Description:

Weave two vectors together, alternating between one letter of the first vector and one letter of the second vector, such that the output vector contains the characters of the first vector at its odd indices and of the second vector at its even indices.

Example:

Given the character vectors `'ILv ALB'` and `'oeMTA!'`, your function should output the vector `'I Love MATLAB!'`

Notes:

- You are guaranteed that the input vectors will be of the same length ($N \geq 1$).
- Be sure that your output is of type `char`.

Hints:

- Think about how you can use the colon operator to create an index vector.

Function Name: airContour

Inputs:

1. (*double*) A 1xN vector representing the pressures of warm air
2. (*double*) Another 1xN vector representing the pressures of cold air

Outputs:

1. (*logical*) Whether or not there will be a change in weather

Background:

Weather changes can occur as a result of collisions between warm and cold air masses. As a meteorologist, you have measured the varying pressures of two air masses, one warm and one cold. As the two collide, you want to determine if there will be a change in weather.

Function Description:

Before you can compare the two vectors, you must first find the contour of each. The contour of a vector describes whether each transition between adjacent elements is increasing or decreasing. If the two vectors have the same contour, meaning both their pressures increase and decrease at the same points, then there will not be a change in weather, as the warm and cold pressure changes will cancel each other out. For the contours 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. The function should output false if and only if the two vectors have the same contour.

Example:

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]. Thus, there will not be a change in weather, and the output would be false.

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 is [increasing, decreasing]. Here the output would be true.

Notes:

- Both vectors will always be the same length.
- Adjacent elements will never have the same value, i.e. there will never be a change of 0.
- 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 `isequal()` functions will be useful.
- If N is a number, the sign of N can be found using the formula: $sign\ N = \frac{N}{|N|}$

- You can use the ~ operator to negate a logical.

Function Name: forecast

Inputs:

1. (*char*) A character vector of the current forecast
2. (*char*) The word to be replaced
3. (*char*) The new word to put into the forecast

Outputs:

1. (*char*) The updated forecast description

Banned Functions:

strrep()

Background:

You have recently started your internship as a weather engineer at the local news station. Unhappy with how the current forecasts are described as 'gloomy' and 'cold', you decide to use MATLAB to replace these words with some happier words!

Function Description:

Write a function that takes an input string, a word to be replaced, and the word to replace it with. Your function should output the original string, but with the one word corrected to the new word.

Example:

```
[out] = forecast('This snowy weather is so cold.','cold','awesome')
```

```
out => 'This snowy weather is so awesome.'
```

Notes:

- The word to be replaced is guaranteed to appear in the forecast string, but only once.

Hints:

- strfind() will be useful in solving this problem.

Function Name: coldWar

Inputs:

1. (*char*) a scrambled character vector
2. (*double*) an integer representing the shift of odd indices
3. (*double*) an integer representing the shift of even indices

Outputs:

1. (*char*) the unscrambled character vector

Background:

While enjoying the winter weather with some pals, you decide to have a snowball fight. Your teammate, on the other side of Tech Green, wants to communicate a frosty message to you without the opposing side being able to decode it. They decide to shout a scrambled version of your message along with two numbers that represent shift values. Since your and your friend are well versed in MATLAB, you decide to write a quick function to decode the encrypted message.

Function Description:

You must 'shift' the letters at even indices by a certain amount and the letters at odd indices by a different amount. For example, if the letter is 'a' and the shift amount is 4 the letter will then become 'e'. A negative shift means shifting towards 'a' in the alphabet. The shift should 'wrap' around the alphabet - 'x' shifted by 7 should become 'e'.

Example:

```
[out] = coldWar('usph',2,-4)
out => 'word'
```

Notes:

- There will be no spaces or uppercase letters.
- The shift may be greater than 26. A positive shift of 28 is the same as a shift of 2.

Hints:

- `mod()` will be useful.