Drill Problem #1

Function Name: socialMedia

Inputs:

1. (*struct*) A 1x4 structure array containing fieldnames and data

Outputs:

1. (string) Social media to which you are most addicted

Function Description:

Since coming to college, you have noticed that you are getting more and more addicted to social media. You decide that you are going to pick the social media that you are most addicted to and quit it cold turkey. To figure out which one to quit, you decided to write a MATLAB function.

This function will take in a structure array, with fieldnames `‘Facebook’`, `‘Twitter’`, and `‘Instagram’`. In each category you will be given how many times you post in one day, how many likes you get on each post, how many comments you get on each post, and how much time you spend on each app per day, in that order. Thus, you will have an input structure array for which the 4 structures correspond to posts, likes, comments, and time, and each of these structures has the fields `‘Facebook’`, `‘Twitter’`, and `‘Instagram’`.

To figure out which one you are most addicted to, use the following algorithm:

$$\text{Addiction score} = \frac{(number\ of\ posts)\times(time\ spent)}{(number\ of\ likes)+(number\ of\ comments)}$$

Once you determine your addiction score for each social media, the one with the highest score is the one you are the most addicted to; output the name of this app as the final output.

Notes:

- The different apps are not guaranteed to be in the same order every time.
- You are guaranteed to not have a tie.

Drill Problem #2

Function Name: walmart

Inputs:

1. (*struct*) A 1XM structure array with fields ‘Item’ and ‘Cost’
2. (*cell*) An Nx2 cell array with item names in the first column and their costs in the second column

Outputs:

1. (*struct*) The updated structure array with the item names alphabetized and the replacement costs implemented

Function Description:

You are a manager at your local Walmart and have just recently started experiencing a massive decline in sales. To your dismay, you discover that a Target has opened up down the street and is selling several items at very competitive prices. Determined to prove yourself as the manager of the best general goods retailer in town, you’ve compiled a list of the items that have a lower price tag at Target, as well as the price that’s beating yours out. All you have to do now is update your own item list so you can match Walmart’s prices to those of Target. That’d be a lot of work to do by hand, but, luckily, you can write a MATLAB function to do this for you!

Write a function that takes in a structure array with fields ‘Item’ and ‘Cost’ as well as an Nx2 cell array containing a list of the items that are cheaper at Target in the first column and their respective costs from Target in the second column. First, sort the structure array in alphabetical order by item name. Then, go through the structure array and, if the item is present in the cell array, replace the cost in the structure array with the associated cost in the cell array.

Notes:

- It may be useful to check how the `sort()` function works on cell arrays.
- The prices listed in the cell array are guaranteed to be lower than the ones listed in the original structure.
- The list of Target’s items is not guaranteed to be in alphabetical order.
- The structure of Walmart’s items and the cell array of Target’s items are not guaranteed to be the same length.

Drill Problem #3

Function Name: myStruct

Inputs:

1. (*cell*) A 1xN cell array containing fieldnames and data

Outputs:

1. (*struct*) The contents of the input cell array in the form of a structure

Function Description:

Write a Matlab function that creates a structure from a given cell array. The cell array will be formatted in the following way:

```
inCellArr   =   {<field1_name>,   {cell_array_of_field1_contents},
<field2_name>,   {cell_array_of_field2_contents},   <field3_name>,
{cell_array_of_field3_contents}, <field4_name>, ...} %etc.
```

Take the cell array and create a structure array, where the fieldnames are given by the strings in the odd-numbered indices and the field contents are given by the field content cell arrays in the even-numbered indices. Each element in the field content cells should be stored individually in the corresponding index of the structure (i.e. if the field content cell arrays are 1xM, then the output structure should be 1xM). The field content cell arrays will all be the same size, with the singular exception of a 1x1 cell. In this case, that single element should be assigned to that field name for every index of the structure (i.e. if your structure is 1xM, then all M elements for this field will be the same piece of data, which is the contents of the singular cell).

Notes:

- The input cell array will always have an even length.
- **You may NOT use the `struct()` function**. The premise of this problem is for you to thoroughly learn how it works by coding it yourself. Any submission that uses the `struct()` function will receive 0 credit.
- You must always assign the contents of a 1x1 cell array into each index of the output structure, not the 1x1 cell itself.

Drill Problem #4

Function Name: omgShoes

Inputs:

1. (*struct*) An MxN structure array representing a shoe store's inventory.
2. (*struct*) A 1xL structure array representing one day's orders.

Outputs:

1. (*struct*) An MxN structure array representing the same store's updated inventory.

Function Description:

You're working at a shoe store back in the year 2007, when the relatively new phenomenon of YouTube viral videos is gripping the country. Everyone who comes into your store thinks they're original when they make an "OMG Shoes…" joke. After two days you can't stand it anymore and you give up your coveted position working on the floor to stay in the back and keep track of inventory, where at least you won't have to hear the same bad joke over and over again. And now you can write a MATLAB function to do your job for you!

Given two structures, one representing the store's inventory and another representing a day's worth of orders, write a MATLAB function to update a store's inventory. The first input will be the inventory, an MxN structure array with fields for `Model`, `Sizes`, and `Stock`:

- `Model`: This field contains a string of the model shoe.
- `Sizes`: This field contains a vector of the available sizes of each model shoe.
- `Stock`: This field contains a vector for each model shoe. This vector corresponds to the Sizes vector, and tells how many of each size is in stock.

The second input to the function will be the orders structure array, which will be a 1xL structure array with fields for `Model`, and `Size`:

- `Model`: This field will contain a string of which model shoe was ordered and will match one of the model shoes in the store's inventory.
- `Size`: This field will contain a single number, which will be the size shoe that was ordered.

To update the store's inventory find each model shoe and subtract 1 from the stock of whichever size was ordered. If an order is the last stock of that particular size and model, then remove that size from the `Sizes` vector and `Stock` vector of that shoe. If an order is the last stock of the last available size of a model shoe, then you should set *both* the `Sizes` and `Stock` vectors of that model shoe to the string '`Out of Stock`'.

Notes:

- You are guaranteed that every model shoe in the order (second) input will exist in the store's inventory (first) input.
- You are guaranteed that every shoe ordered will initially be in stock.

Drill Problem #5

Function Name: ministryOfMagic

Inputs:

1. (*char*) Name of .txt file containing a description of a dark wizard
2. (*struct*) Array of aurors and stats

Outputs:

1. (*char*) Statement of battle outcome for the Daily Prophet

Function Description:

Dark wizards are at large in the Wizarding World and you are the Head Auror in the Ministry of Magic. For those of you who do not have an inherent love of Harry Potter, this means that criminals are at large and you're the chief of police in the local governmental organization. Your job is to choose which auror in your task force to dispatch based on how powerful the dark wizard is.

The first line of the dark wizard text file will be the name of the dark wizard in question. First, you must count the number of threat words contained in the input dark wizard text file (the threat words are `‘dark’`, `‘lord’`, `‘cruciatus’` and `‘death’` and it does not matter what case they are in). To determine how powerful the dark wizard is, **you must double the total number of threat words** in the text file. Then, you must determine which auror is best suited to fight the dark wizard. The structure array of aurors has the fields `‘name’`, `‘experience’` (number of years they've worked), `‘strength’` (on a scale from 1-5), and `‘rank’` (on a scale from 1-5). To calculate how powerful an auror is, you must use the following formula:

$$power = strength + rank + \frac{experience}{2}$$

You must choose an auror that is more powerful than the dark wizard; however, since you do not want to expend a high level auror on a low level dark wizard, **the auror you select must also be the closest in power to the dark wizard, while still being stronger**.

Finally, you must submit a statement to the Daily Prophet (the newspaper) saying `'The Ministry of Magic Department of Magical Law Enforcement dispatched <auror_name> to take on <dark_wizard_name>. <auror_name> succeeded in apprehending <dark_wizard_name>.'`

If two aurors have the same level of power, choose the one that appears earlier in the structure array (for example, if the two that are the same are at indices 3 and 5, choose number 3). If no aurors are qualified to fight the dark wizard, select the strongest auror on the task force and then submit the statement `'The Ministry of Magic Department of Magical Law Enforcement dispatched <auror_name> to take on <dark_wizard_name>. <auror_name> failed to apprehend <dark_wizard_name>.'`

Finally, if the dark wizard is Voldemort and Harry Potter is one of the aurors in the structure array, you must select Harry Potter and output the statement `‘The Ministry of Magic Department of Magical Law Enforcement dispatched Harry Potter to take on Voldemort. Harry Potter absolutely destroyed Voldemort.’`

Good luck, Head Auror!