

Homework #06: Conditionals

Drill Problem #1

Function Name: criticalFailure

Inputs:

1. (*double*) A 1x2 vector of numbers from 1-20, inclusive

Outputs:

1. (*char*) An output string describing the result of your die roll

Function Description:

To the tune of "Mr. Brightside" by The Killers

*I'm coming out of my cage
And I've been doing just fine
Gotta Gotta make friends because I want them now
It started out with Catan—
How did it end up like this?
It was only a game, it was only a game*

*Now I'm casting some spells
And she's rolling a die
While he's having a snack and she's failing her saves
Now they're going to fight
And I'm under-leveled
And it's all in our heads, but she's stealing his*

*Loot now
He rolls for perception
Let me go
I just can't stop playing D&D
It's taking control*

Write a MATLAB function to emulate a die roll for an attack in a Dungeons and Dragons game you are playing. If the first roll (the first number in the vector) is anything other than a 1, output 'Good Job!'. If the roll is a 1, however, you have critically goofed, and a second die will be rolled (the second number in the vector) to determine what damage you may have caused to yourself. The table below shows what your output should be for each roll of the second die:

Die Roll	Function Output
1-5	'Right Leg Wounded'
6-10	'Left Leg Wounded'
11-15	'Right Arm Wounded'
16-20	'Left Arm Wounded'

Homework #06: Conditionals

Drill Problem #2

Function Name: foolproof

Inputs:

1. (*double, char, or logical*) An input that could be correctly or incorrectly formatted

Outputs:

1. (*char or logical*) A message prompting a different input, or the output of `powerOfTwo()`

Function Description:

At your first day of work, your boss asked you to write a MATLAB function to determine whether a number is a power of two (it will play a vitally important role in an initiative to aid in the global integration of synergy between interdepartmental paradigms). Conveniently, you wrote a function to do exactly this back in CS 1371. After checking the code with a few test cases, you upload the function to your department's shared folder. Within minutes your coworkers have flooded your email with complaints about errors in your code; they tried inputting decimals, logicals, and strings—but the function did not work like it was supposed to.

You realize there is only one option to fix this: write a MATLAB function called `foolproof()` to ensure that an input is correctly formatted before it is called in `powerOfTwo()`. The input for `powerOfTwo()` should be a natural number (a natural number is a positive integer; zero is not included for the purposes of this problem). If an incorrect input is given, output one of the following messages:

1. If the input is of type double but it is not a natural number, then the output should be:
`Please input a natural number.`
2. If the input is of type logical, then the output should be:
`The input type is invalid. Please input a natural number.`
3. If the input is of type char, then the output should be:
`When you use the ' symbol, it clogs the gears in the computer's calculator. Please input a natural number.`

Given a correctly formatted input, assign the output of `powerOfTwo()` to this function's output.

Notes:

- The input can be of type `'logical'`, `'char'`, or `'double'`.
- The strings above appear as they would in the Command Window. They will need to be formatted differently being assigned to a variable to achieve this format.
- You may want to use your `powerOfTwo()` function from previous weeks as a helper function (the solution code will be posted Tuesday for it).
- Your string must EXACTLY match the output string. The `isequal()` function may be useful.

Hints:

- The functions `isnumeric()`, `islogical()`, and `ischar()` may be useful.

Homework #06: Conditionals

Drill Problem #3

Function Name: rPS

Inputs:

1. (*char*) Player 1's name and move (format: 'Name-Move')
2. (*char*) Player 2's name and move (format: 'Name-Move')

Outputs:

1. (*char*) Sentence containing outcome explanation and winning player(s)

Function Description:

Want to play Rock, Paper, Scissors? No thanks! The old R-P-S game is so boring and traditional. Instead of plain-old rock, how about some crystals with personalities, like timeless quartz, or chalky limestone? You decide to make a new R-P-S game that allows users to add panache to their moves. Write a MATLAB function that determines which player wins this new spin on Rock, Paper, Scissors.

The rules are as follows:

All Rock Types Are Equal:

Possible Rock Types: 'Sapphire', 'Ruby', 'Quartz', 'Limestone', 'Diamond'.

All Scissor Types Are Equal:

Possible Scissor Types: 'Safety', 'Edge', 'Hair', 'Craft'.

All Paper Types Are Equal:

Possible Paper Types: 'Cardstock', 'Origami', 'Wrapping', 'Graphing'.

	Possible Scenarios	
	When Someone Wins	When a Tie Occurs
	Paper Type > Rock Type	Rock Type == Rock Type (Tie)
	Rock Type > Scissor Type	Scissor Type == Scissor Type (Tie)
	Scissor Type > Paper Type	Paper Type == Paper Type (Tie)
Function Output	'<P# move> beats <P~# move>. <P# name> wins.' (where # is the player with better move)	'Tie; try again.'

Notes:

- All names and moves will be capitalized and separated by a hyphen.

Hints:

- Switch statements may be very useful.

Homework #06: Conditionals

Drill Problem #4

Function Name: `slidinDemDMs`

Inputs:

1. (*char*) An input string representing a message from your romantic interest

Outputs:

1. (*char*) An output string describing your result

Function Description:

As you are paying close attention to your professor in your CS 1371 lecture, you are distracted by a couple in the row in front of you. There is some serious flirtation going on, and you find yourself thinking, “Hey, I want what they have!” The only problem is that while it is easy to see and identify someone else trying to flirt, you are completely lost as to whether the person you flirt with is flirting back or not. Luckily, as a quickly burgeoning programming aficionado, you decide to employ MATLAB to help you out.

Write a function in MATLAB called `slidinDemDMs()` that will take in a string representing a text or other direct message from your crush and output a description of his or her interest. You will do this by calculating a score for the message based on the frequency of specific characters in the message as follows:

Character	Points
' :) ' or ' (: '	5
' ;) ' or ' (; '	10
' <3 '	15
' ! '	2
'bae '	5
'Heyy '	2 (+2 for each additional 'y')

You will then output one of the following results based on the total score of your input:

Score	Function Output
0-14	'You are looking for love in all the wrong places.'
15-29	'Like a ball at the top of a hill, you have potential.'
30+	'Are you a bank? Because you have acquired interest!'

For example, the input `'I <3 MATLAB!!!! :)'` from your crush would get a score of 30 and the output would be `'Are you a bank? Because you have acquired interest!'`. All words will be separated by spaces.

Notes:

- Keywords `'bae '` and `'Heyy '` can be capitalized or lowercase—account for both.
- You will only have one instance of the `'Heyy '` pattern in any given input string.
- This program has a less than 100% effectiveness rate in real life—tread lightly.

Hints:

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

Drill Problem #5

Function Name: battleRoyal**Inputs:**

1. (*char*) A string containing the names of three wrestlers, separated by commas
2. (*double*) A 1x3 vector of numbers 0-100 matching the strengths of each wrestler
3. (*logical*) A 1x3 logical vector indicating if each wrestler is a face

Outputs:

1. (*char*) An output string describing the outcome of the match

Function Description:

Welcome to the exciting world of WWE, where professional wrestlers duke it out in action-packed fights to determine who will be the next World Champion. One such fight is the battle royal, in which 20 male wrestlers enter the ring, but only one leaves victorious. As a professional wrestling enthusiast, you wish to predict who this winner will be. Since you were busy doing Drill Problems 1 - 4, you missed the majority of the fight, and now only 3 wrestlers remain in the ring.

Each of these wrestlers has a unique name given in a comma separated string, as well as a given score based on his overall strength. This score is always a whole number between 0 and 100. The higher a wrestler's score, the stronger he is in the ring. Lastly, each wrestler can either be described as a face (good guy) or heel (bad guy). Faces tend to work together in the ring against heels, and vice versa. You can account for this behavior by manipulating the wrestlers' strength scores. If, out of the remaining 3 contestants, there are 2 faces and only 1 heel, then the heel's strength score should decrease by 3 points. On the other hand, if there are 2 heels against 1 face, the face's strength should decrease by 3 points. Finally, if the remaining 3 wrestlers are all faces or all heels, then all of their scores should remain the same.

Given this information, write a function in MATLAB named `battleRoyal()` that takes in these three inputs and outputs a string describing the outcome of the battle royal. The different outcomes are as follows:

- The strongest and second strongest wrestlers have the same strength scores; they will eliminate each other, and the third strongest wrestler will win. In this case, the output string should read `'The underdog <name> wins!'`
- The difference between the first and second strongest wrestlers is only 1 or 2; the strongest wrestler will win. In this case, the output string should read `'<name> just barely takes the victory!'`
- The difference between the first and second strongest wrestlers is 3 or 4; the strongest wrestler will win. In this case, the output string should read `'The champion is <name>!'`
- The difference is 5 or greater between the strongest wrestler and either of the others; the strongest wrestler will win. In this case, the output string should read `'<name> wins by a landslide!'`

Homework #06: Conditionals

Drill Problem #5

- o Lastly, if 'John Cena' is one of the names in the first input, then you should completely ignore the above guidelines and automatically output 'John Cena wins by absolute domination!!!'

Where the <name> variable above is replaced by the corresponding wrestler's name, taken from the first input.

For an example, given the three inputs:

```
str = 'DM Punk, Triple H-B, McDaniel Bryan'
strengths = [100, 94, 93]
faceHeel = [true, true, true]
```

Because all three of these wrestlers are faces, we don't have to manipulate their strengths. We can then conclude that 'DM Punk' is the strongest, beating out the second strongest ('Triple H-B') by 6 points. Therefore, the output string would read: 'DM Punk wins by a landslide!'

Notes:

- It is guaranteed that all three wrestlers will never have the same strength, so you do not have to worry about a tie scenario.

Hints:

- The `strtok()`, `sort()`, and `sprintf()` functions may be useful.