

Function Name: earthWindFire

Inputs:

1. (*char*) Player 1's move
2. (*char*) Player 2's move

Outputs:

1. (*char*) A string describing who won

Function Description:

You and your friend are listening to some music while working on your CS 1371 homework when you realize that September is ending. As passionate music enthusiasts, you quickly engage in an argument debating who would win in a battle between Earth, Wind, and Fire. You invent a game based on Rock, Paper, Scissors and start competing with each other using parts of your favorite band's name.

1. Earth beats wind.
2. Wind beats fire.
3. Fire beats earth.

Based on these rules, the function should output one of the following strings:

```
'Player 1 wins!' if player 1 wins  
'Player 2 wins!' if player 2 wins  
'It's a tie!' if both players choose the same move
```

Notes:

- If you didn't catch the reference, check out this [song](#).
- You are guaranteed that both of the inputs will be lowercase and one of the 3 possible moves.

Function Name: wakeMeUp

Inputs:

1. (*char*) The time when you wake up on a 24-hour scale (format: '<hour>:<minute>')
2. (*char*) The time of your first class
3. (*char*) Mode of transport to get to your first class

Outputs:

1. (*char*) A string describing whether or not you were late and why

Function Description:

As you hear the sound of your roommate's alarm going off, playing their favorite Green Day song, "Wake Me Up When September Ends", you bolt upright in bed in a panic. You didn't set your alarm! Will you be late to class?? Use MATLAB to determine the fate of your morning attendance by following these rules:

1. If you oversleep and wake up after or at the exact time of your first class, output this:
'You overslept and missed your first class. Better set your alarm next time!'
2. If you wake up before your first class, there's hope you might make it! Consider your mode of transportation to see if you will make it to class this morning.

Mode of Transport	Number of minutes
'skateboarding'	3
'walking'	10
'stinger bus'	15

Output this if your transportation gets you to class after it starts:

'You rush to get to class by <mode of transport>, but sadly you are still late. Better set your alarm next time!'

Or this if you get there exactly at the right time or before your class starts:

'You rush and make it to class just in time by <mode of transport>. Nice!'

Notes:

- All times will be written on a 24 hour scale. So 1:40 pm would be '13:40'.

Hints:

- You can use `strtok()` to separate a string based on any delimiting character.
- Use `%s` for adding a string variable when using `sprintf()`.
- Consider using a switch statement when dealing with the mode of transportation.

Function Name: highSchoolMusical

Inputs:

1. (*char*) Applicant's dream university
2. (*double*) 1x3 vector of applicant's scores in singing, dancing, & acting abilities (in order)
3. (*char*) Applicant's song they will sing at the audition
4. (*logical*) Logical indicating whether applicant's headshot is a selfie (true indicates selfie)

Outputs:

1. (*char*) String indicating whether or not the applicant got a part in the musical

Function Description:

Your high school is having a musical, and they put you in charge of casting! But who will you cast as the lead in the play? How will you compare all the different attributes of each applicant? Then you realize, this is the perfect opportunity for MATLAB! Write a function that takes in various attributes of an applicant and outputs a string of whether or not they got the part.

To calculate the scores, you decide to weight the singing, dancing, and acting scores differently. Singing is weighted at 150%, dancing is weighted at 125%, and acting stays the same (100%). Use the following rules to determine if they get the part:

1. They must get at least 26 total points (weighted) to get the part.
2. The applicant can sing any song, but if it's by 'Nickelback', they don't get the part.
3. If they have a selfie as their headshot, they are not professional enough to get the part.

However, there are some exceptions.

- If the applicant's dream school is 'Georgia Tech', then they may have a selfie and a score less 26 points. But, they still cannot sing a 'Nickelback' song.
- If the applicant sings your favorite song, 'Start Of Something New', then they automatically get the part.
- Finally, you're calling it all off if the applicant's dream school is 'u(sic)ga'.

Even if they have the best scores or sing your favorite song, you would never let a dawg into your precious musical.

If the applicant is successful and earns the part in the musical, output this statement:

'Welcome to High School Musical!'

If the applicant does not get the part in the musical, output this statement:

'We regret to inform you that you did not make the musical.'

Notes:

- 'Georgia Tech', 'GT', or 'Georgia Institute of Technology' are all valid inputs.
- u(sic)ga will be written only as 'u(sic)ga'.
- If the song is by Nickelback, 'Nickelback' will appear somewhere in the song input.
- It doesn't matter if your favorite song is capitalized or not, you'll still love it.

Hints:

- A switch statement may be helpful!

Function Name: findMyJam

Inputs:

1. (*char*) Comma separated list of two potential artists to play
2. (*char*) Comma separated list of your friends' favorite artists
3. (*char*) Comma separated list of the top 5 Spotify artists of the week
4. (*double*) 5x2 array of stats for the top 5 Spotify artists of the week

Outputs:

1. (*char*) A string describing your final choice of which artist to play

Background:

You and your friends have been busy working on your MATLAB homework all week, but it's finally Friday and you have some time to relax and unwind. To celebrate finishing your homework, you decide to throw a pizza party for your friends, but you can't decide what music to play that will get everyone excited. You come up with two potential artists, but want to get the opinions of your friends, along with information on what's popular on Spotify this week. With this information, you can figure out which artist to play at your party.

Function Description:

The first 3 inputs to your function will be strings in the following formats:

```
'<artistOption1>,<artistOption2>',  
'<friendsArtist1>,<friendsArtist2>,<friendsArtist3>',  
'<spotifyArtist1>,<spotifyArtist2>,<spotifyArtist3>,<spotifyArtist4>,<spotifyArtist5>'
```

The fourth input of your function will be a 5x2 array of stats, with each row corresponding to a specific artist and the columns corresponding to either total plays that week or rating, in the following format:

```
      total plays  rating  
stats = [ 1500000    4.3; <- spotifyArtist1  
          75000     3.7; <- spotifyArtist2  
          45000     4.9; <- spotifyArtist3  
          780000    2.7; <- spotifyArtist4  
           4000     1.5 ] <- spotifyArtist5
```

You decide on the following set of rules:

1. If artist option 1 is in the friends list and in the Spotify top artists list, play artist option 1.
2. If artist option 1 is in only the Spotify top artists list, but not the friends list:
 - a. And if artist option 1 has a Spotify rating greater than the average rating of the top 5 artists, play artist option 1.
 - b. If not, play the top rated artist in the Spotify list.
3. If artist option 1 is only in the friends list, but not in the Spotify list:
 - a. And if artist option 2 is in the friends list and in the Spotify list, play artist option 2.
 - b. Otherwise, play artist option 1

Continued...

4. If artist option 1 is in neither the friends list nor the Spotify list:
 - a. And artist option 2 is in either the friends list or the Spotify list, play artist option 2.
 - b. Otherwise, play the top rated artist in the Spotify list.

Your output string should be in the format:

```
'<artistName> will ignite the party!'
```

Notes:

- There will be only single commas dividing artist names.
- All artist ratings will be positive numbers.

Hints:

- Consider how you can relate the row index in the stats array to the respective artist in the Spotify list by comma position.
- The `strfind()` and `find()` functions may prove useful.

Function Name: musicalYahtzee

Inputs:

1. (*char*) A 1xN string of notes shown on the die after rolling

Outputs:

1. (*double*) The maximum number of points possible
2. (*char*) A string recommending the corresponding dice combination

Background:

You and your roommates are going to a concert of your favorite band this weekend and you need to brush up on your knowledge of musical notation, so you decide to play a game of Musical Yahtzee. All the rules of Yahtzee apply, but instead of the numbers 1 to 6 on the die, there are the Thirty-second notes, Sixteenth notes, Eighth notes, Quarter notes, Half notes, and Whole notes. Since you are a Tech student, your competitive nature drives you to write a function in MATLAB to make sure you find the best combination in a set of five notes.

Function Description:

You will first need to convert the input string, which is comprised of words separated by spaces (representing the note on each of the 5 dice rolls), into a 1x5 vector of integers 1 to 6, as in a traditional Yahtzee game. Use this table to convert the notes to numerical values:

Note Name	Appearance in Input String	Corresponding Numerical Value
Thirty-second Note	'thirty-second'	1
Sixteenth Note	'sixteenth'	2
Eighth Note	'eighth'	3
Quarter Note	'quarter'	4
Half Note	'half'	5
Whole Note	'whole'	6

For example, if given the input:

```
diceNotes = 'half quarter thirty-second eighth half',
```

the corresponding numerical vector would be:

```
vecNotes = [5 4 1 3 5]
```

Continued...

Homework 6 - Conditionals

The dice combinations are as follows:

Combination	Description	Points
'3 of a Kind'	One note appears 3 times.	Sum of all dice
'4 of a Kind'	One note appears 4 times.	Sum of all dice
'Full House'	3 of one note and 2 of another note	25
'Small Straight'	4 consecutive notes	30
'Large Straight'	5 consecutive notes	40
'Yahtzee'	All five dice are the same note	50
'Chance'	Any other sequence	Sum of all dice

Determine which combination(s) your dice fall under. If you dice meet multiple criteria, you need to determine which combination scores the most points. Your first output should be the maximum score for the given input. Your second output should be a string with the following format identifying which combination is being used to score the points:

```
'I have a <combination>!'
```

If your combination is a Chance, you must have had bad luck rolling your dice. Replace the '!' with a space and a sad face as follows:

```
'I have a Chance :('
```

However, if your combination is a Yahtzee, your entire second output should simply be:

```
'HOORAY! MUSICAL YAHTZEE!'
```

Notes:

- While 'Small Straight' and 'Large Straight' require sequential numbers to be present, these numbers may not be in order. For instance, the vector [4 1 3 5 2] is a 'Large Straight'.

Hints:

- You may find the `sort()` and `diff()` functions useful.
- You may find a helper function to be useful in solving this problem.
- Be careful about the order of your conditionals.