

**Function Name:** boxOfficePower

**Inputs:**

1. (*char*) Comma separated string of three possible actors for your superhero movie

**Outputs:**

1. (*double*) Your movie's projected earnings at the box office (in millions)

**Background:**

You've just made a fortune as a MATLAB coder and have moved to L.A. to become a hotshot Hollywood producer. As you look at the current state of Hollywood movies, you notice one thing: the box office is dominated by franchises. Ever so eager to get in on the fun (and money), you decide to start your own superhero franchise!

**Function Description:**

For your superhero movie to be successful, it needs actors with lots of box office draw. Each actor has the potential to bring in a certain sized fanbase based on their appeal. You decide to write a MATLAB function that takes in a list of three potential actors and calculates your movie's gross earnings (in millions) based on its cast. The list of actors will be a character vector separated by commas. To calculate your total projected earnings, add together the box office draws of each individual actor. Use the following table:

Actor	Box Office Draw (Millions \$)
Gal Gadot	1,900
Robert Downey Jr.	800
Chadwick Boseman	400
Chris Hemsworth	350
Scarlett Johansson	300
Jeremy Renner	150
Brandon Routh	50
Ryan Reynolds	0
Nicolas Cage	-200

*Continued...*

**Example:**

```
>> earnings = boxOfficePower('Chris Hemsworth,Chadwick Boseman,...  
Jeremy Renner')  
  
earnings => 900
```

**Notes:**

- Actors will be separated by one comma and one comma only.
- You can assume that actors will only show up once.
- The movie cast will only contain the actors listed above.

**Hints:**

- A helper function may be useful.
- Consider which type of conditional would be more convenient to use here.

**Function Name:** civilWar

**Inputs:**

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

**Outputs:**

1. (*char*) A statement describing which player won

**Background:**

You and your friends decide to play rock-paper-scissors, but immediately get bored. Then you remember Marvel's new movie *Black Panther* is almost in theaters and decide to spruce up the classic game to include your favorite Avengers. Once you agree upon the rules, you write a MATLAB function to determine the winner of each round.

**Function Description:**

Write a function that takes in and compares two of the following strings. These are the **ONLY** inputs we will use to test your function:

'Captain America'  
'Black Widow'  
'Hawkeye'  
'Black Panther'  
'Iron Man'

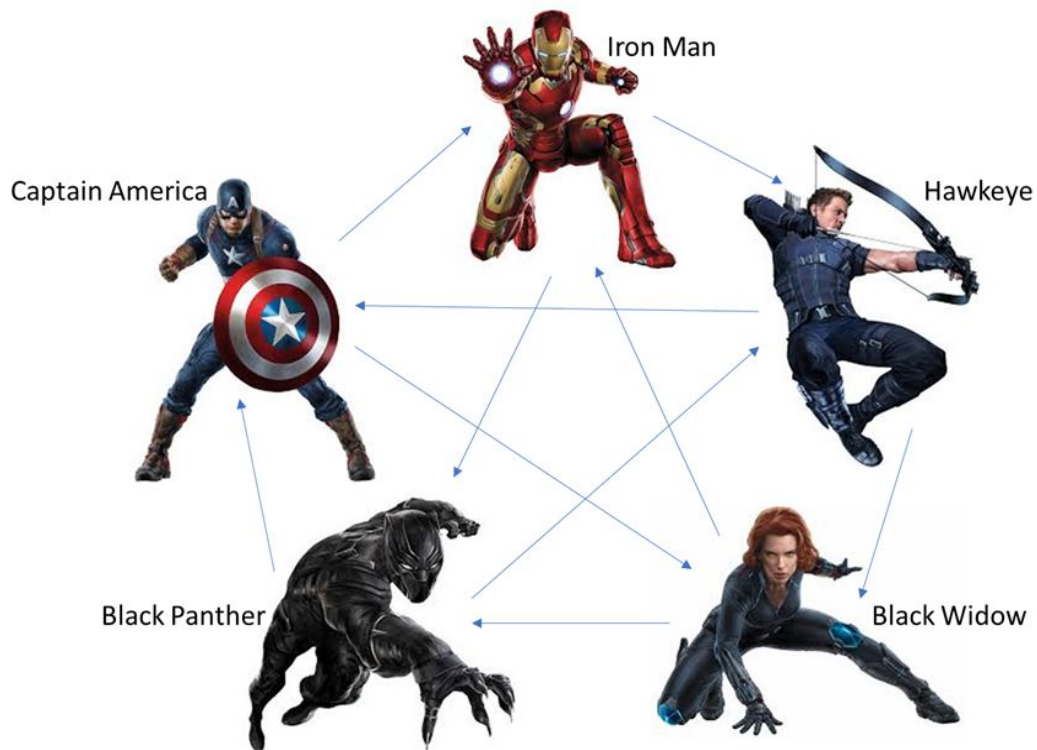
Then compare them by the following rules (look on the following page for a graphic):

1. Black Panther beats both Captain America and Hawkeye.
2. Captain America beats both Iron Man and Black Widow.
3. Iron Man beats both Black Panther and Hawkeye.
4. Hawkeye beats both Captain America and Black Widow.
5. Black Widow beats both Black Panther and Iron Man.

It 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

*Continued...*



**Example:**

```
>> winner = civilWar('Captain America', 'Black Panther')
```

```
winner =  
    'Player 2 wins!'
```

**Hints:**

- There are only 25 possible test cases. If you match the solution for all test cases, you are guaranteed to get all the points for this problem.

**Function Name:** incredibleUno

**Inputs:**

1. (*char*) The name of the current player
2. (*char*) A letter representing the color of the card on the top of the discard pile
3. (*double*) A number representing the value of the card on the top of the discard pile
4. (*char*) A 1xM character vector representing the colors of the cards in the player's hand
5. (*double*) A 1xM vector representing the values of the cards in the player's hand

**Outputs:**

1. (*char*) A description of what was played

**Background:**

The Incredibles are playing Uno for their family game night. In Uno, the point of the game is to be the first person that runs out of cards, and luckily for these heroes, one of MATLAB's superpowers is the ability to find the best moves to make in one play.

**Function Description:**

Given the color and number of the card on top of the discard, write a function that finds the best way to get rid of the most cards in your player's hand and outputs the move.

Here are the rules of this Uno game:

- The next played card must match the top card's color or number.
- Multiple cards can be played if and only if they have the same number as the top card.
- Draw from the pile if nothing can be played. When 'Frozone' can't play, he will forfeit the game and get up to leave, saying 'Where's my super suit?!'

Based on the rules and your inputs, you will output one of the following strings:

Scenario	Output
Played number card(s)	'<player name> played <# of cards played> #<card number> card(s).'
Played color card	'<player name> played 1 <card color> card.'
Had to Draw a Card	'<player name> had to draw a card...'
Frozone Exception	'Where's my super suit?!'

*Continued...*

**Example:**

```
[out1] = incredibleUno('Mr. Incredible', 'r', 8, 'rbgyb', [7 8 6 8 3])  
>> out1 = 'Mr. Incredible played 2 #8 cards.'
```

**Notes:**

- The possible card colors are as follows:
  - red --- 'r'
  - blue --- 'b'
  - yellow --- 'y'
  - green --- 'g'
- Since the goal is to play as many cards as possible, prioritize having the same number as the top card before having a card with the same color.
- If a player plays one card, the last word of the output string should be 'card', but if the player plays more than one card, the last word should be 'cards'.

**Hints:**

- The second output for the `max()` function might be helpful.

**Function Name:** beMyHero

**Inputs:**

1. (*char*) Comma separated list of 2 superhero movie options to play
2. (*char*) Comma separated list of your friends' favorite superhero movies
3. (*char*) Comma separated list of the top 5 superhero movies from Netflix
4. (*double*) A 5x2 array of stats for the top 5 superhero movies from Netflix

**Outputs:**

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

**Background:**

It's Friday! You and your friends have a movie night in your dorm. But you all can not settle on what movie to watch that will get everyone excited. You come up with two Marvel movie options, but decide to consult your friends' opinion and Netflix movie ratings. With this information, you decide to use MATLAB to figure out which movie to play at your movie night.

**Function Description:**

Your function will have the following three inputs in the following order and format:

```
'<movieOption1>,<movieOption2>'
'<friendsMovie1>,<friendsMovie2>,<friendsMovie3>,...,<friendsMovieN>'
'<Netflix1>,<Netflix2>,<Netflix3>,<Netflix4>,<Netflix5>'
```

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

	<i>total revenue</i>	<i>rating</i>	
stats = [	1500000	4.3;	<- Netflix1
	75000	3.7;	<- Netflix2
	45000	4.9;	<- Netflix3
	780000	2.7;	<- Netflix4
	4000	1.5 ]	<- Netflix5

You decide on the following set of rules:

1. If movie option 1 is in both lists, play movie option 1.
2. If movie option 1 is in only the Netflix list, but not the friends list:
  - a. And if movie option 1 has a Netflix rating greater than the average rating of the top 5 movies, play movie option 1.
  - b. If not, play the top rated movie in the Netflix list.
3. If movie option 1 is in only the friends list, but not in the Netflix list:

*Continued...*

- a. And if movie option 2 is in the both lists, play movie option 2
  - b. Otherwise, play movie option 1
4. If movie option 1 is in neither list:
  - a. And movie option 2 is in either list, play movie option 2
  - b. Otherwise, play the top rated movie in the Netflix list.

Your function should output a string in the following format:

`'Based on condition <num>, <movieName> would be a great choice!'`

The `<num>` is a number, 1-4, depending on the conditions specified in the rules above.

**Notes:**

- There will be only single commas dividing movie names.
- All movie ratings will be positive numbers.
- There can be any number of movies in your friends' favorite movie list.

**Hints:**

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



**Function Name:** heroMask

**Inputs:**

1. (*char*) Filename of an image of a superhero without his or her mask, including the file extension

**Outputs:**

None

**File Outputs:**

1. Image of the superhero with the proper mask applied

**Background:**

One of the greatest challenges of being a superhero is protecting your identity while fighting crime and saving the day. Several of our favorite heroes have been spotted in their suits without their masks on! You must use your knowledge of MATLAB to identify the superhero in each image and cover their face with a picture of their masked face before the images reach the public!

**Function Description:**

First, we must determine which of the masks to apply to the picture of the superhero. To do this, find the average R, G, and B values of the image (find the average values of the red layer, blue layer, and green layer). The proper mask can be found using the ranges provided in the table below:

	Red	Green	Blue
<b>mask1</b>	(60,100]	(60,100]	(60,100]
<b>mask2</b>	[0,60]	[0,60]	[0,60]
<b>mask3</b>	[0,60]	(60,100]	[0,60]
<b>mask4</b>	(100,255]	(100,255]	(100,255]
<b>mask5</b>	(60,100]	[0,60]	[0,60]

\*Note that ( ) indicate non-inclusion and [ ] indicate inclusion, so (60,100] means  $60 < x \leq 100$ .

The mask images are stored in filenames 'mask<#>.png'. These images are the same size as their corresponding background image, and contain the hero's mask, with a pure green (RGB = [0, 255, 0]) background.

Create a new image where the superhero's face is covered with the appropriate mask and save this image with the name '<superhero>\_wMask.png', where <superhero> is the base of the input filename.

*Continued...*

**Example:**

```
'theflash.png'
```

```
=>
```



```
heroMask('theflash.png')
```

```
=>
```

```
'theflash_wMask.png'=>
```

**Notes:**

- There will be some green leftover around the edges when you apply the mask, as seen above. This is all right, as long as your output matches that of the solution function. You can use the provided `checkImage()` function to test your outputs. Type `help checkImage` in the command window for more information on how to use it.
- It is guaranteed that the averages will be in one of the specified ranges.
- The solution file will create an updated image titled '`<superhero>_wMask_soln.png`'.