

Drill Problem #1

Function Name: howSingle

Inputs:

1. (*char*) A string of your major at Georgia Tech
2. (*char*) A string of your gender
3. (*double*) A number of which year you are in school
4. (*logical*) Whether or not you still have all your first gen. holographic pokémon cards

Outputs:

1. (*char*) A description of how single you are

Function Description:

It's Valentine's Day Week here at Georgia Tech, and, because of this, students all over campus are re-evaluating their love lives. Unfortunately, we Tech students are just too busy to really get out there and start dating, so we want a guarantee that our time will be well spent. Luckily, you can write a MATLAB function to tell you just how single someone really is (or is not!) and whether or not they should go out there and mingle or settle in for a night of singleness and start a new series on Netflix. All MATLAB needs is someone's major, year, and gender to tell them just how single they really are!

Here's how it works: each person receives points depending on their input major, with 'Business' receiving one point, 'Engineer' receiving three points, 'CS' receiving five points, and anything else receiving zero points for this category. However, if they are any of the first three majors and still have all of their first gen. holographic pokémon cards, they will receive an additional 15 points!

In terms of student gender, 'Male' students receive one point and 'Female' or other students receive 3 points. Finally, students receive one point to their overall score for each year they are in school.

To calculate the student's overall score, simply add the points they received for their major, gender, and year, and if the score is ten or above then MATLAB will output 'You're totally HOT, you won't be single long!' (which will appear in the Command Window as `You're totally HOT, you won't be single long!`). If you get a score between five and ten, MATLAB will output 'You're not completely undateable, keep trying!' (which will appear in the Command Window as `You're not completely undateable, keep trying!`). Finally, if your score is below five MATLAB will output 'You might want to renew your Netflix subscription.' (which will appear in the Command Window as `You might want to renew your Netflix subscription.`). Now get to writing! It's time to figure out just how single you are!

Notes:

- This is not an accurate representation of your love life!! All of you are perfect. ☺

Function Name: rPSL**Inputs:**

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

Outputs:

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

Function Description:

Rock, paper, scissors...love? A fun twist on a classic game, R-P-S-L is played using the normal Rock-Paper-Scissors rules with the addition of an all-powerful new sign: Love. The rules are as follows:

Paper>Rock	Rock=Rock (Tie)	Love>Rock/Paper/Scissors
Rock>Scissors	Scissors=Scissors (Tie)	Love==Love (Win/Win)
Scissors>Paper	Paper=Paper (Tie)	

Given two strings containing players' names and moves, write a function to determine who wins a game of R-P-S-L. Outputs should be formatted as follows:

Love/Love:	'Love conquers all! <P1 name> and <P2 name> both win.'
Love/RPS:	'Love conquers all! <P# name> wins.' (# player who picked 'Love')
RPS Tie:	'Tie, try again.'
RPS/RPS:	'<P# move> beats <P~# move>. <P# name> wins.' (# player with better move)

Notes:

- All names and moves will be capitalized and separated by a hyphen.
- In the event of a regular RPS/RPS win, the second move should appear in the output as lowercase (ex: 'Rock beats scissors.').

Function Name: loveCalculator

Inputs:

1. (*char*) string representing a name
2. (*char*) string representing another name

Outputs:

1. (*double*) calculated love potential score

Function Description:

Contrary to popular belief (and to Shakespeare), a rose by any other name would not smell as sweet or be loved as much. Love is not determined by arbitrary things like astrological signs, appearances, shared traits, or even personalities. Love is determined using a basic algorithm that compares two names. In honor of Valentine's Day, your task is to help your peers by writing a function named "loveCalculator" that takes in two strings representing the names of two people and calculates their love potential using the algorithm that determines love.

This algorithm consist of two parts:

First, if the two names are identical, or if one of the names is in the other (e.g. 'alexa' and 'alexander', 'jordan' and 'dan'), the love potential score is zero. If the two names are not identical or contained within each other, then move onto the second part.

For this second part, calculate the love potential by counting the number of occurrences of the chars 'l', 'o', 'v', 'e', and 's' present in both names. Then put these digits together to create a new number and add each digit within the number to its adjacent digit/digits.

For the example, if given the names 'ophelia' and 'hamlet', counting each letter results in:

'l' = 2 'o' = 1 'v' = 0 'e' = 2 's' = 0 which makes the total number 21020.

Each digit of this number is added to the adjacent digit or digits to form new totals.

'l' + 'o'	'o' + 'v'	'v' + 'e'	'e' + 's'
2+1 = 3	1+0 = 1	0+2 = 2	2+0 = 2

These totals are again arranged as digits of a single number (new number is now 3122).

This process is repeated until only two numbers remain. This is the final love potential score.

3+1 = 4 1+2 = 3 2+2 = 4 which makes the new total number 434

4+3 = 7 3+4 = 7 which makes the final love potential score 77

Notes:

- The names are guaranteed to be all lowercase letters.

Hints:

- You may find the functions `strfind()`, `str2num()`, and `num2str()` helpful.

Function Name: aragornAndArwen

Inputs:

1. (*char*) A string of a type of creature
2. (*double*) A number that represents facial hair length
3. (*logical*) A 1x3 logical vector representing the possession of three various kinds of power: [high social status, physical strength, possession of magical ability]
4. (*double*) A number representing an age

Outputs:

1. (*char*) The result of matchmaking process

Function Description:

The One Ring has been destroyed and there is finally peace in Middle Earth. All the creatures of Middle Earth look at Aragorn and Arwen's beautiful relationship and feel envious. They want to find true love, too. So, you decide to write a MATLAB function to help them (yes, MATLAB and computers existed in Middle Earth. You just never noticed because you were too focused on Orlando Bloom playing Legolas).

Your function will take in 4 inputs. The first is a string with the type of creature. This can be a 'dwarf', 'elf', 'hobbit', 'man', 'orc', or 'Gollum'. The second is a number ranging from 1 to 4 that correlates to the type of facial hair on the creature. The third is a logical vector that contains `true` and `false` responses to questions about high social status, physical strength, and possession of magical ability. For example, if the creature is a strong king, the input would be [true true false]. The fourth input is a number for the age of the creature.

If the creature is an orc, your function will output the result 'Your only shot at love would be a match made in Mordor.'. If the creature is Gollum, your function will output the result 'You lost your Precious, the One Ring. Please, try again later.'. If the creature is anything else, the Middle Earth Matchmaker gets cranking. Each creature can receive a total score out of 80, with 20 possible points based on each input of the function:

1. If the creature is an elf, he is awarded 30 points out of 20 because elves are the coolest creatures in Middle Earth, obviously. All other creatures are awarded 20 points out of 20.
2. Concerning facial hair, this input number will be an integer ranging from 1 to 4. 1 representing no facial hair, 2 representing stubble, 3 representing a short beard, and 4 representing a long beard. Dwarves are awarded points in increments of 5 for increasing hair length. This means that a dwarf with no facial hair is awarded 5 points and a dwarf with a long beard is awarded 20. Men are awarded 20 points for having stubble and 10 points for all other cases. Elves and hobbits are awarded 20 points only if they have no facial hair and 0 points under all other circumstances.
3. In relation to power, if at least 2 of the statements are true, the creature is awarded 20 points. If only 1 statement is true, the creature is awarded 10 points and if none of the statements are true, the creature is not awarded any points.

Homework #6: Conditionals

Drill Problem #4

4. Finally, age is dependent on the type of creature. Elves are immortal but only reach maturity at the age of 50, so they are awarded 20 points only if they are at least 50. Dwarves and men and hobbits are awarded 20 points only if they are within the bounds of 20-50% of their average lifespan. Men and hobbits have an average lifespan of 100 years, while dwarves have an average lifespan of 250 years.

After you have found all the points earned for each input, simply add up the points to get the percentage that this specific creature receives out of 80, rounded to the hundredths place. If a creature point total is at least 50 out of 80, they will find a match and the result should be 'You have ____% odds of finding your soul mate in Middle Earth.', where you will print their total point percentage into the string. Otherwise the result should be 'Your only shot at love would be a match made in Mordor.'. Please help the poor, lonely creatures of Middle Earth by coding this MATLAB function. They only want what Aragorn and Arwen have.

Notes

- To get a percent sign into a string using `sprintf()`, you need to type two percent signs into the format string. For example, `sprintf('%%')`, will return the string `'%'`.
- When using `sprintf()`, the `%0.2f` option allows you to print to two decimal places.

Homework #6: Conditionals

Drill Problem #5: Extra Credit

THIS PROBLEM IS EXTRA CREDIT!

Function Name: matchZodiac

Inputs:

1. (*char*) A person's birth date
2. (*char*) Another person's birth date

Outputs:

1. (*char*) The romantic compatibility of the two people

Function Description:

It's Valentine's Day, you're feeling a little cheesy, so you decide to consult the astrological powers of zodiac signs to see if you and your (potential) significant other are truly compatible. Write a function that takes in two people's birth dates, in the format of `'February 14'` (with the month spelled out and a space between the month and day), and outputs their compatibility based on the table below.

Sign	Birthday Range	100%	25%	Wild Card
Aries	Mar 21- Apr 19	Leo, Sagittarius	Libra, Cancer, Capricorn	Pisces, Taurus
Taurus	Apr 20 - May 20	Virgo, Capricorn	Scorpio, Leo, Aquarius	Aries, Gemini
Gemini	May 21 - Jun 20	Libra, Aquarius	Virgo, Pisces, Sagittarius	Taurus, Cancer
Cancer	Jun 21 - Jul 22	Scorpio, Pisces	Capricorn, Libra, Aries	Gemini, Leo
Leo	Jul 23 - Aug 22	Aries, Sagittarius	Aquarius, Scorpio, Taurus	Cancer, Virgo
Virgo	Aug 23 - Sep 22	Taurus, Capricorn	Pisces, Gemini, Sagittarius	Leo, Libra
Libra	Sep 23 - Oct 22	Gemini, Aquarius	Aries, Capricorn, Cancer	Virgo, Scorpio
Scorpio	Oct 23 - Nov 21	Cancer, Pisces	Taurus, Leo, Aquarius	Libra, Sagittarius
Sagittarius	Nov 22- Dec 21	Aries, Leo	Gemini, Virgo, Pisces	Scorpio, Capricorn
Capricorn	Dec 22 - Jan 19	Taurus, Virgo	Cancer, Aries, Libra	Sagittarius, Aquarius
Aquarius	Jan 20 - Feb 18	Gemini, Libra	Taurus, Scorpio, Leo	Capricorn, Pisces
Pisces	Feb 19 - Mar 20	Cancer, Scorpio	Gemini, Sagittarius, Virgo	Aries, Aquarius

For instance, if the two inputs were `'March 21'` (Aries) and `'July 24'` (Leo), then the function should output `'100%'`. For matches that are not covered by the table, such as `'December 21'` (Sagittarius) and `'February 18'` (Aquarius), the function should output `'50%'`. The function will have four possible outputs: `'100%'`, `'25%'`, `'Wild Card'`, and `'50%'`. However, keep in mind that in real life, two people of any birth sign can have a healthy and long-lasting relationship.

Notes:

- Leap years (years with a February 29th) should be taken into account. That is, you should approach the problem with 366 days in a year in mind.

Hints:

- The Zodiac compatibility table follows a certain pattern.