

dFunction Name: `magneticFlux`

Inputs:

1. (*double*) The value of R (the wire resistance)
2. (*double*) The value of L (the side length of the square wire)
3. (*double*) The value of I (the current)

Outputs:

1. (*double*) Changing magnetic field

Topics: (*mathematical formulas*), (*trigonometry*)

Background:

George P. Burdell was super happy last week when he solved for the magnetic field of a triangular wire in CS1371. However, he just realized he can change his code to solve one of his physics 2 HW problems!

Function Description:

Given R , L , and I , calculate the rate at which the magnetic field is changing (your output). Use the following equation to solve for changing the magnetic field: dB/dt

$$V = \frac{dB}{dt} A$$

You can calculate the values of V and A through the following equations:

$$V = IR$$

$$A = L^2$$

Finally, round your answer to 2 decimal places.

Example:

```
sideLength = 2;  
resistance = 5;  
current = 1;
```

```
dBdt = magneticFlux(resistance, sideLength, current);
```

```
dBdt → 2.89
```

Function Name: `shiftCode`

Inputs:

1. (*double*) An integer

Outputs:

1. (*char*) The uppercase letter of the alphabet corresponding to the input integer.

Topics: (*mod*), (*ASCII values*), (*casting*)

Background:

George P. Burdell is competing in a Tech themed scavenger hunt during his first week on campus, and is stuck on a clue that requires him to convert numbers to letters. In classic engineer fashion, he turns to MATLAB to help decipher the code and identify the letters needed for this clue!

Function Description:

Write a function that converts the input number to its corresponding uppercase letter. 'A' maps to 1, and 'Z' maps to 26. If the number is beyond this range, it should wrap back to the beginning. For example, 27 would correspond to 'A', and 28 would correspond to 'B'.

Example:

```
>> out1 = shiftCode(1)
out1 → 'A'
```

```
>> out2 = shiftCode(22)
out2 → 'V'
```

```
>> out3 = shiftCode(34)
out3 → 'H'
```

Notes:

- All inputs will be positive integers.

Hints:

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

Function Name: `charSplit`

Inputs:

1. (*char*) A decimal number

Outputs:

1. (*char*) The left letter of the split
2. (*char*) The right letter of the split

Topics: (*rounding*), (*ASCII math*), (*string-number conversions*)

Background:

George P. Burdell is working on a new decoration for his residence hall! He is looking to hang each letter of the alphabet up along the top of the hallway, but he's got a pesky corner that will cause his alphabet to be split up onto two different walls and is unsure of which letters should be divided by the corner! Luckily, you live just down the hall and can use your new MATLAB skills to help him out!

Function Description:

You are given a string that holds a decimal number, describing which letter will split the alphabet. If your input is a decimal number, then that means your alphabet will be split between two letters. Your first output will be the letter to the left of the split and your second output will be the letter to the right of the split. If your input is a whole number, then your split occurs on a letter and you should output the same letter for both outputs.

Example:

```
[leftLet, rightLet] = charSplit('13.3')
leftLet → 'M'
rightLet → 'N'

[leftLet, rightLet] = charSplit('21')
leftLet → 'U'
rightLet → 'U'
```

Notes:

- Your alphabet should be uppercase letters only.
- The letter 'A' will be described by the number 1 and the letter 'Z' will be described by the number 26.
- All inputs will be integers between 1 and 26 (inclusive).

Hints:

- The `str2num()`, `floor()`, and `ceil()` functions may be useful.
- Don't forget about doing math with ASCII values.

Function Name: `bandsawScraps()`

Inputs:

1. (*double*) The amount of inches of wood you want to cut off

Outputs:

1. (*double*) The amount of feet of wood that will be left
2. (*double*) The amount of inches of wood that will be left

Topics: (*mod*), (*functions*)

Background:

George P. Burdell needs to cut wood for his ME 2110 robot. He knows how much wood he wants to cut-off but also wants to know how much wood will be left. This seems like a perfect task for MATLAB!

Function Description:

Given the amount of inches of wood to cut off, calculate the amount of wood that will be left in feet (first output) and inches (second output). Assume that you will always start with 10 feet of wood. Notice that there are 12 inches in a foot.

Example:

```
inchesToCut = 88;
[feetLeft, inchesLeft] = bandsawScraps(inchesToCut)
feetLeft → 2
inchesLeft → 8
```

Notes:

- All inputs will be positive integers less than or equal to 120 inches
- Your second output must be less than 12 inches.

Extra Credit

Function Name: `napTime`

Inputs:

1. (*double*) The current position of the hour hand
2. (*double*) The current position of the minute hand
3. (*double*) A positive or negative number of minutes

Outputs:

1. (*double*) The position of the hour hand after the specified time
2. (*double*) The position of the minute hand after the specified time

Topics: (*mod*)

Background:

George P. Burdell has had quite a busy first week! He has homework to do, but right now he just needs to take a nap. He knows what time it is right now and he is trying to figure out what time it will be if he naps for a given amount of minutes. This sounds like a problem MATLAB could solve!

Function Description:

This function will take in the current position of the hour hand, as an integer between 0 and 11 (0 for noon/midnight), the current position of the minute hand, as an integer between 0 and 59 (0 for "on-the-hour") and a positive or negative number of minutes elapsed.

Given this information, determine the new position of the clock hands. You should assume that the hour hand does not move until the next hour has begun. For example, the hour hand stays on 2 from 2:00 until 2:59 and only at 3:00 does the hour hand move to 3.

Example

```
[hour, min] = napTime(11, 30, 122);  
hour → 1  
min → 32
```

Notes:

- The `mod()` and `floor()` functions will be useful.
- As you do this problem, notice the behavior of `mod()` when the first input is negative.

Hints:

- Information on using `mod` in this manner:
 - [Khan Academy](#) - [Better Explained](#)