Function Name: `freefall`

Inputs:

1. *(double)* The initial velocity of an object in m/s
2. *(double)* An amount of time in seconds

Outputs:

1. *(double)* The final velocity of the object after the given time

Function Description:

Write a MATLAB function that will calculate the velocity of an object in freefall given an initial velocity (in m/s) and an elapsed time in seconds since the initial velocity. Assume that the object is in free fall toward Earth, so the acceleration due to gravity is 9.807 m/s^2. You can also neglect terminal velocity, air resistance or any other factor not taken into consideration by the following kinematic equation:

$$v_f = v_i + a * t$$

where v_f is the final velocity, v_i is the initial velocity, a is the acceleration of the object, and t is the elapsed time. Assume that a negative initial velocity means it is traveling upwards; positive means it is traveling downwards.

Function Name: `splitTicket`

Inputs:

1. *(double)* The total cost of the meal
2. *(double)* The tip percentage expressed as a decimal
3. *(double)* The number of people in the party

Outputs:

1. *(double)* The amount each person has to pay

Function Description

It's never fun doing the math to calculate the tip at a meal. It's even less fun having to do the math to split the ticket. However, you will be the coolest person around if you whip out MATLAB Mobile and run this function to do it all for you! The function will take in the total cost of the meal, the amount you wish to tip (expressed as a decimal) and the number of people in your party. It will output the amount each person should pay (to the nearest cent) assuming you split the bill evenly.

Function Name: `fib`

Inputs:

1. *(double)* Which Fibonacci number to calculate

Outputs:

1. *(double)* The specified Fibonacci number

Function Description:

You are probably familiar with the Fibonacci sequence: the sequence of numbers (starting with 1, 1) where every number is the sum of the previous 2 numbers. From this definition, you might think in order to calculate the nth Fibonacci number, you would have to first calculate all the numbers that came before it. However, this is not the case! The formula for any Fibonacci number is:

$$F_n = \frac{\phi^n - (-\phi)^{-n}}{\sqrt{5}}$$

φ is the golden ratio:

$$\phi = \frac{1+\sqrt{5}}{2}$$

To calculate any Fibonacci number, all you have to do is plug ‘n’ into this equation.

This formula is mathematically exact (no rounding is necessary in theory) but because of MATLAB’s internal rounding of large numbers, you should round your answer to the nearest integer.

Function Name: `triArea`

Inputs:

1. *(double)* One of a triangle's angles in degrees
2. *(double)* Another of the triangle's angles in degrees
3. *(double)* The length of the base of the triangle

Output:

1. *(double)* The area of the triangle

Function Description

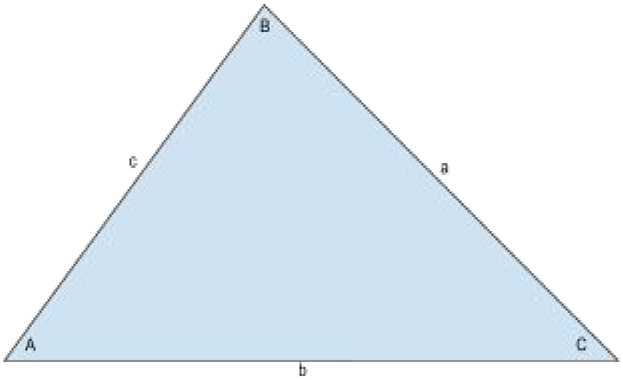
Given the base of a triangle (the third input) and the two angles on either side of the base (the first two inputs) calculate the area of the triangle. There are many different formulas you could use to solve this problem, but the most direct way is to use the Law of Sines to calculate another side length, then use the following area formula:

$$Area = \frac{a*b*sin(C)}{2}$$

The Law of Sines states that:

$$\frac{sin(A)}{a} = \frac{sin(B)}{b} = \frac{sin(C)}{c}$$

All variables are in terms of the following triangle:



Your answer should be rounded to the hundredth place.

Notes:

- Remember to use `sind()` instead of `sin()` when doing trig calculations in degrees.